Mission Cockpit

Handbook

Version 0.7.0

0. Contents

1. General Information

Before Mission Cockpit can be used, some preparations must be carried out. This may take some time. Take the time or let it remain the same ;-) But believe me, the time spent, is worth it!

So please read the following important points to understand and conscientiously implement!

Use at your own risk!
No guarantee on accuracy, completeness or function.
No liability for any direct or indirect personal or property damage.

Note the rules for the operation of radio controlled model aircraft!

Note the license terms for Mission Cockpit in Chapter 23.

2. Updating from previous program versions

Settings are stored in the XML file **`mkCockpit.xml`**. With newer versions of the program configuration entries are added, for which there is normally no default settings, but which are absolutely required by the program. Therefore, in case of an existing **`mkCockpit.xml`**, then data must be manually merged with the program attached **`mkCockpit.xml`** for the respective version.

This can be done with any editor, it is comfortable with WinMerge http://winmerge.org/

3. Perl Interpreter

Mission Cockpit is programmed in the scripting language **Perl/Tk**. To run the program, the **ActivePerl Community Edition version 5.10**, 32 **bit X86** is needed**.** Newer versions of Perl than 5.10 or the 64 bit version was still not suitable because the TK or Win32**:** API is not available for this. The Perl interpreter for Windows can be found here*:* http://www.activestate.com/activeperl/downloads

Mission Cockpit requires a number of **additional Perl packages**. To install these packages, the script **`InstallPackages.bat`** from subdirectory **`Perl`** must be run. The script downloads the required packages from the Internet, and installs them**.** Also, when updating from previous versions of Mission Cockpit, you should let this script run**.**

Mission Cockpit is not available as an .EXE file.

4. Supported Operating Systems

The program has been developed to run under Windows 7 (32 +64 **bit)**, Windows **Vista and Windows XP**.

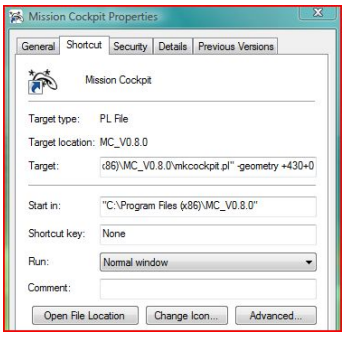
5. Starting the Program

From the CMD shell with the command:

```
mkcockpit.pl
```

or:

```
perl mkcockpit.pl
```



Of course, you can also create a shortcut on the desktop, e.g.:

Hint:
After starting the program, must be at least six satellites received, before the position of MK is updated on the map.

Command Line Parameters	Description
`-geometry <value>`	Position and size of screen window, eg: +430+0 to establish the position on screen.
`-<section>:<key> <value>`	Thus the parameters of **`mkcockpit.xml`** can be overriden.

6. Flight-Ctrl and Navi-Ctrl Prerequisites

The following software versions match:

FC	NC	Mission Cockpit
0.73	0.14	0.1.x
0.74	0.15	0.2.x
0.76	0.17	0.3.0
0.78	0.18	0.4.x
0.80	0.20	0.5.x
0.82	0.22	0.6.x
0.84	0.24	0.7.x
0.86	0.26	0.8.x

7. Create a Map Definition

7.1. Airfield - Background Image

The map of the airfield must be available as JPEG or PNG file. You can have any size**,** but should completely fit on the screen**.** Required data is stored in the map**.** The map can have any orientation; the North must not be at the top**.** As source can be for example any screenshot of Google Earth or GeoMapTool

7.2. Airfield - Map Definition

Directory map can hold multiple files with map definitions for several airfields. The selection of appropriate map takes place in the Mission Cockpit configuration dialog:

`File → Preferences → Map→ Map default Background Image.`

For historical reasons, there are four alternative methods to create a map definition:

7.2.1. Download maps from OpenStreetMap

This variation is supported as of version Mission Cockpit 0.5.2**.** It is now possible to create directly from the Mission Cockpit a calibrated map**.** The program uses map data from OpenStreetMap and thereby position and compass data from MK**.**

Important:

Note the terms of use, of the map provider:

OpenStreetMap: http://www.openstreetmap.org/ FAQ Terms of use

Method**:**

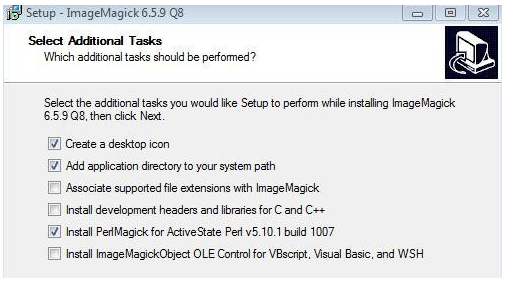
- **Installing ImageMagick**

  For image processing, the program will require ImageMagick. During the installation of ImageMagick is also needed a Perl package **`Image::Magick`** to be installed with.

Important:

In the Install dialog, must necessarily be selected:

`Install PerlMagick for ActiveState Perl V5.10.1 build 1007`

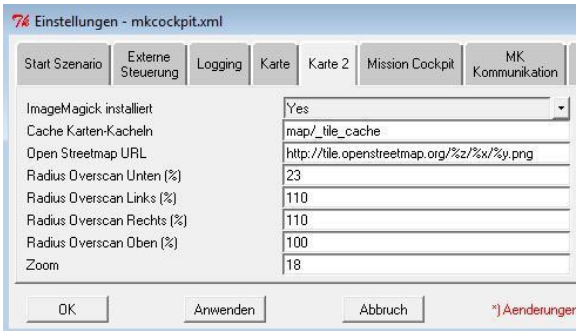


A matching version to **`Perl 5.10.1 Build 1007/1008`** ImageMagick can be found here:
http://image_magick.veidrodis.com/image_magick/binaries/ImageMagick-6.5.9-9-Q8-windows-dll.exe

Important:

The contained in the ImageMagick Perl package **`Image::Magick`** is strongly depended to the installed Perl version.
For the above mentioned Active State Perl 5.10.1**,** the above mentioned ImageMagick version must be used!
If other versions of Perl are to be used, then look for suitable versions of ImageMagick at the directory**:**
http://image_magick.veidrodis.com/image_magick/binaries

- **Activate the download function**



The shipped configuration has the download function initially disabled.

The activation takes place in Preferences dialog.

`ImageMagick installiert` → `Yes`

Important:

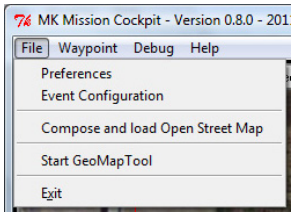
After changing the setting a program restart is required.

Important:

If you activate this feature and ImageMagick is not installed, then Mission Cockpit cannot be started!
In case of an accidental activation this can be undone by editing the mkcockpit.xml file**:**
Change: **`ImageMagickInstalled="Yes"`** to: **`ImageMagickInstalled="No"`**

- **Create a map**

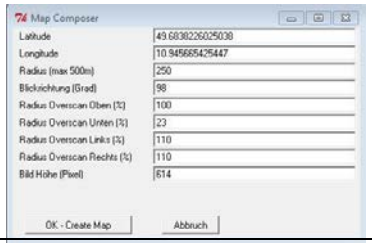
Place MK to the starting position on the airfield and wait for a GPS position fix**,** then select the desired function in the file dialog:



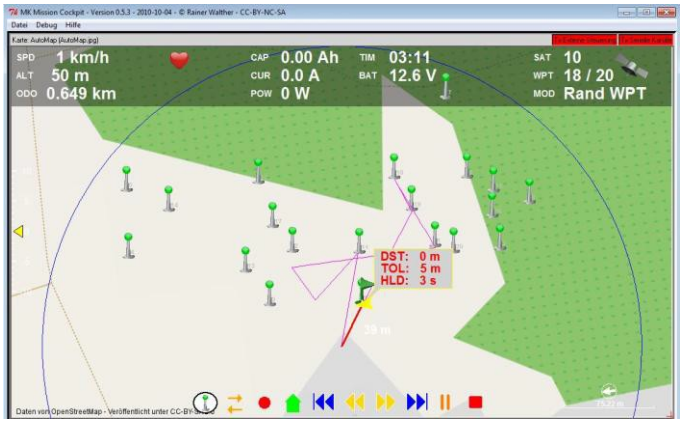
Important:

The menu item appears only if, **`ImageMagick installed`** is activated.

Mission cockpit now reads the following data from MK:

Current GPS-Position	Is used as a card-*center*
Current Compass value	Determines the rotation of the map. The MK then looks at the map up.
Operational radius (GPS max. Radius) of the MK Setting	Determines the shown on the map area.
The height of the screen	Determines the size (height) of the created JPEG File. It proposes 80% of the screen height.
	The calculated values are displayed and can be adapted.

With the parameter **`Radius Overscan xxx (%)`** you can select the desired map section.
The area shown on the map is derived from the operation radius (parameter in the MK-setting) multiplied by the specified percentage of overscan.
With the checkbox **`Reload Tile`** you can force the already existing tiles to be loaded again from the tile server.



The data from the above screenshot for example resulting this map section.

The position of the starting square is moved downward. Below is only 23% of the visible radius. Left and right 110% respectively. Above 100%.

This results in approximately one Aspect ratio of 16:9.

The card is saved with the filename AutoMap.jpg in the map directory.
An existing file is overwritten without asking

- **Internet connection – Cache**

  To download a map an internet connection is required. The map tiles are saved in the cache directory **`map/_tile_cache.`** When creating a map, system checks whether the tile is already in the cache. If it exists in the cache, it is taken from there, and does not download a new from Internet. It does not check if there are more recent data on the Internet
  To force a reload you have to delete the cache directory or manually activate the checkbox:
  `Reload Tile`
  In case of an Internet connection is not available at the airport, you can download at home the required map tiles (enter latitude, longitude, radius in the confirmation screen).

- **Internet connection – Proxy**

  If your Internet access is via a proxy, you must use the **`set HTTP_proxy`** environment variable, e.g.:
  `set HTTP_proxy=http://proxy:8080`

7.2.2. JPEG/EXIF Map Definition from GeoMapTool

This variation is supported as of version Mission Cockpit 0.5.1**.** The required metadata for the map calibration are extracted from the EXIF Comment of the JPEG file. Is required only the JPEG file that is copied into the map directory

The JPEG file can be created easily with the GeoMapTool: http://www.geomaptool.de

When saving in GeoMapTool there are several possibilities, all of which are supported by the Mission Cockpit.

- JPEG for Mission Cockpit, supported from version 0.5.1

  Only JPEG file is required that contains information about:

  - ✓ Map calibration data, even for rotated maps.
  - ✓ Home and POI Position.
  - ✓ Airfield Area Polygon Boundary

- JPEG and XML for Mission Cockpit, supported by all versions.

  This is functionally identical to the first variant. The metadata, however, rather than be embedded to the JPEG file are transferred by a separate XML file. See also Section 7.2.4.

- JPEG of MK Kopter Tool supported from version 0.5.1

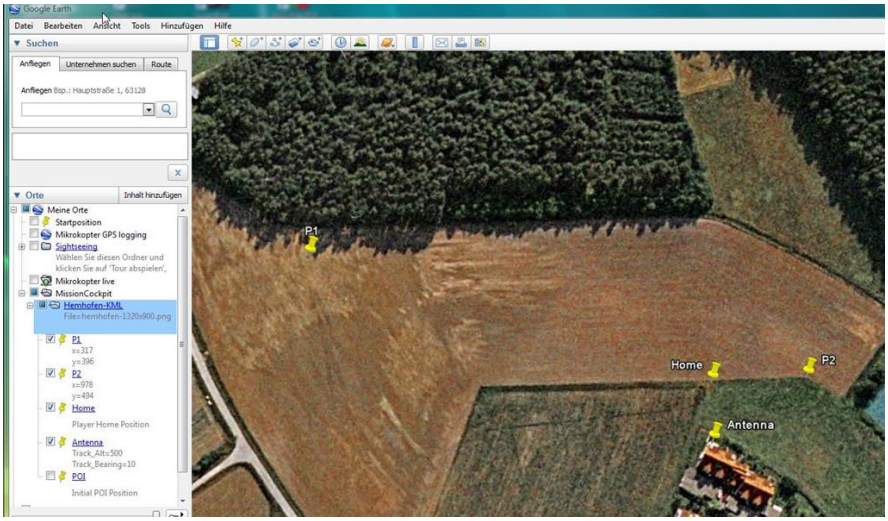
  Only contains the card map calibration. Rotated Map Tiles are not supported. North is always on top.

7.2.3. KML Map Definition

The definition of the cards is created directly in Google Earth and exported from there as a KML file to the **`map`** directory of the Mission Cockpit. This method requires a little manual work and should therefore only be used if it no longer possible with the GeoMapTool or the automatic mapping.

Important:

The file extension of the map definitions file is **`.kml`**



In the area My Places, create a folder with the name of the map definition, e.g.:

`Hemhofen KML`

Attention:
Spaces or special characters are not allowed!

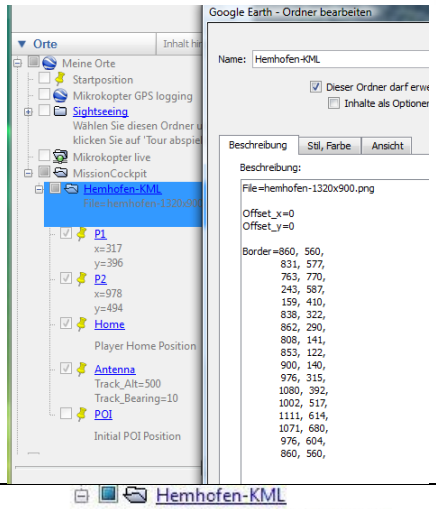
This is the name of the map, that later will be indicated in **Missions Cockpit** configuration dialog.

In the folder's Properties (right click on the folder → Properties) enter the description of the required additional information.

Mandatory entry:
`File=<Filename of Background Image >`

Important:
Pay attention to upper / lower case!

Other parameters are optional, e.g. Airfield Boundary.



Under the folder create placemarks for the two calibration points **P1**, **P2**, and optionally for the **Home**, **Antenna** and **POI** location.

Important:
The placemarks must have exactly this name.

Important:
Pay attention to Upper / lower case

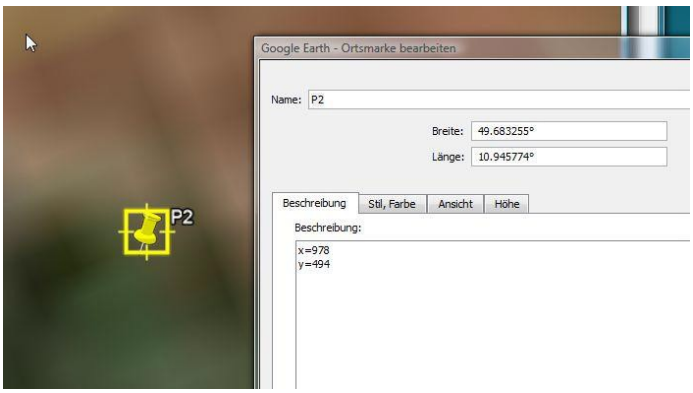
If you zoom in strongly enough in Google Earth, then the pin of the placemark can be positioned very precisely. The needle tip of the pin determines the desired location. The GPS coordinates of the placemark from GE is automatically determined and recorded

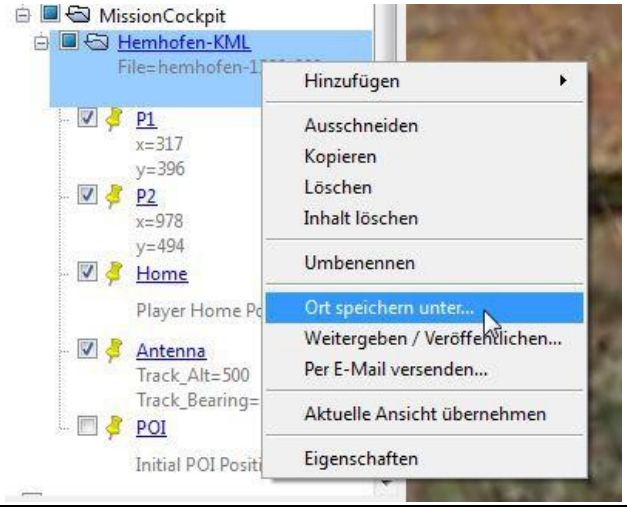
After right clicking on the placemark, you can move the pin of the placemark and modify the properties of the placemark.

The pixel that is needed for the calibration of the map coordinates of the airfield map are entered in the description of the placemark, e.g.

`x = 123`
`y = 456`

Important:
Pay attention to Upper / lower case



Here you can first enter any coordinates. Later during calibration, you can enter the correct coordinates. **Important**: The placemarks with names **P1** and **P2** must be created this way	
Optionally you can create placemarks named **Home**, **POI** and **Antenna**. **Home** and **POI** have no other parameters in the description field. At the placemark antenna direction of the antenna can be optionally specified in the description field of the height of the GPS and compass for example; **`Track_Alt=500`** **`Track_Bearing=10`** **Important**: Pay attention to Upper / lower case	
If the visibility button is disabled, then the placemarks are not considered by Mission Cockpit when data are imported, therefore: **P1** and **P2** must be always visible.	
The folder will then be exported by right-mouse button clicking to the Missions Cockpit **`map`** directory and saved as a **KML** file **Important**: In the file selector dialog box select the **KML** file extension. **KMZ** is active by default. **KMZ** files are not processed by Mission Cockpit. The **KML** file will then be loaded automatically the next time you start the Mission Cockpit and the included map definition can be used. In each **KML** file, only one card definition is allowed	

7.2.4. XML Map Definition

It is the older variant**,** which is much more complicated to use. The XML file must be created manually with a text editor and edited. The file must correspond to the standard XML syntax**.**

Important:
The file extension of the map definition file is. **`.Xml`**.

The XML file has the following structure:

XML-Data	Description			
<mkcockpit-Maps>	Fixed Text			
<Hemhofen	Start a card map definition. Unique name of the card. No spaces or special characters! The XML file can contain multiple definitions.			
Name="Hemhofen"	Unique name of the card. No spaces or special characters are allowed.			
File="hemhofen-800.png"	File name of the background image			
P1_x="66"	Calibration Point	P1	X	Pixel Coordinates
P1_y="62"		P1	Y	
P2_x="778"		P2	X	
P2_y="488"		P2	Y	
P1_Lat="49.685333"		P1	GPS	Latitude
P1_Lon="10.950134"		P1		Longitude
P2_Lat="43.882949"		P2		Latitude
P2_Lon="10.644580"		P2		Longitude
Home_Lat="49.685333"	**Optional:** Home position in the player mode**:** GPS Latitude. If not specified uses the Home position of the MK**.**			
Home_Lon="11.945960"	**Optional:** Home position in the player mode: GPS Longitude. If not specified uses the Home position of the MK**.**			
Track_Lat="48.685333"	**Optional:** Position of the tracking antenna: GPS Latitude. If not specified uses the home position of the MK.			
Track_Lon="11.950134"	**Optional:** Position of the tracking antenna: GPS **Longitude**. If not specified uses the home position of the MK.			
Track_Alt="512"	**Optional:** GPS height of the tracking antenna. If not specified, the amount of the home position of the MK-gyro calibration is used.			
Track_Bearing="10"	**Optional:** Viewing direction of the antenna at the center position. If not specified the value of the Compass MK at the start of the engines is used.			
POI_Lat="48.685333"	**Optional:** Position of the Point of Interest: GPS Latitude.			
POI_Lon="11.685333"	**Optional:** Position of the Point of Interest: GPS Longitude.			
Offset_x="5"	**Optional:** Displacement of objects on the map in the X direction, to compensate for small errors in the calibration.			
Offset_y="5"	**Optional:** Displacement of objects on the map in Y-direction to compensate for small errors in the calibration.			
Border="555,430,516,555, 516,555,516,555,"	**Optional:** The polygon in the x / y pixel coordinates defines the Airfield Boundary. The origin of the coordinate system 0 / 0 is the top left of the map.			
/>	Completion of a map definition			
</mkcockpit-Maps>	End of File data			

8. Map Calibration

The created map cards with the Missions Cockpit integrated card download, or from http://www.geomaptool.de with GeoMapTool, already provide a calibrated map. See also chapters:

7.2.1 Download Maps from OpenStreetMap
7.2.2 JPEG / EXIF Map Definition from GeoMapTool

In this case, no further calibration is required, you can skip this chapter.

Otherwise, the card must be calibrated carefully!
Of these, it depends on how accurately the correct location of the MikroKopter, will be shown on the map

The calibration is done via the two calibration points P1 and P2, for which each of the GPS coordinates (longitude, latitude) and the x/y pixel coordinates of the map must be determined.
P1 and P2 should be as far apart as possible, e.g. P1 top left and P2 bottom right. The best approach is to find two distinct points on the map that can be easily found again.

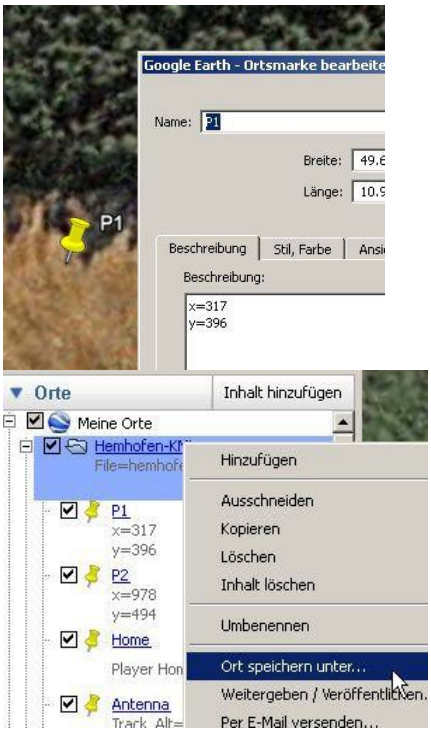
Depending on the type of map definition, KML or XML, the calibration is done differently.

8.1. Calibrate maps with KML definition

As from Chapter 7.2.3 the two placemarks P1 and P2 with provisional x/y pixel coordinates were already defined in Google Earth. We should now obtain the correct x/y pixel coordinates.

For this start Mission Cockpit with the previously stored map definition (File → Settings, Tab: map) and click with the left mouse button on the calibration point location.

The x/y coordinates then are displayed in the status bar and also copied to the Windows Clipboard.



In Google Earth, you add the pixel coordinates in the description field of the corresponding placemark with Ctrl-V. The GPS coordinates (Lat, Lon) are also inserted, which should be cleared away, once again.

Important:
This must be done for both calibration points P1 and P2.

Do not forget:
Then, the folder must once again be saved to the Missions Cockpit map directory as a KML file and the old file that contains still the wrong x/y coordinates to be overwritten.

The calibration is then complete, and the card map definition can be used, after a restart of the Mission Cockpit

8.2. Calibrate maps with XML definition

The GPS coordinates and the matching x / y pixel coordinates must be obtained manually, and then manually entered with an editor in the XML file in the Keys:

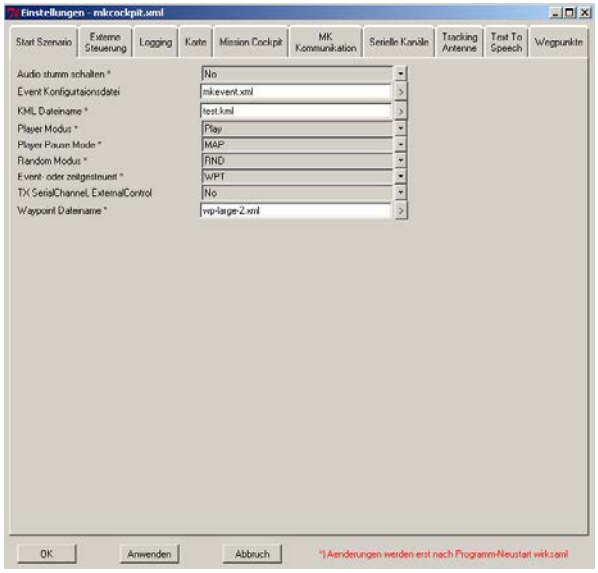
`P1_Lat, P1_Lon, P1_x, P1_y, P2_Lat, P2_Lon, P2_x, P2_y.`

See the description of the XML file format in section 7.2.4

Since this approach is rather cumbersome and error-prone, I will not go further on that matter.

9. Configuration

Mission Cockpit has extensive options for configuration. The configuration is stored in the file **`mkcockpit.xml`**.



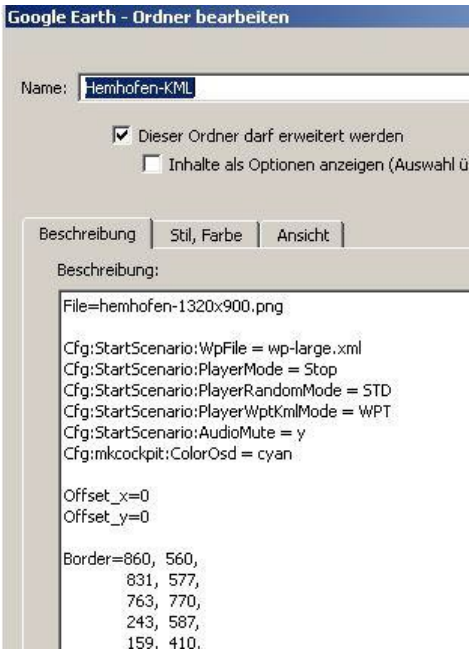
Configuration changes take place at File menu → settings.

Important:
Starting with version 0.5.1, most of the settings in "OK" or "Apply" are take place immediately.
For older versions, the settings don't take effect until you restart the program.

For Experts:
The file mkcockpit.xml can be edited directly using a suitable text editor.
To obtain a good contrast between the card and the objects drawn on the map, it may be necessary to change the color of the represented objects.
Colors can be specified as: Name, such as red green, blue, or RGB hex values, for example: #FF0000 (red)

9.1. Configuration depending on the map-definition

Sometimes it may be necessary to have specific settings for an airfield, e.g. other colors to make the objects stand out better from the map background, or as to start with a different scenario.



Such settings can be stored as of version 0.2.6 directly in the map definition. This works for almost all parameters defined in the **`mkcockpit.xml`**.

This applies with the following syntax:

`Cfg:<Sektion>:<Key> = Value`

`<Sektion>` Section Name
`<Key>` Parameter Name

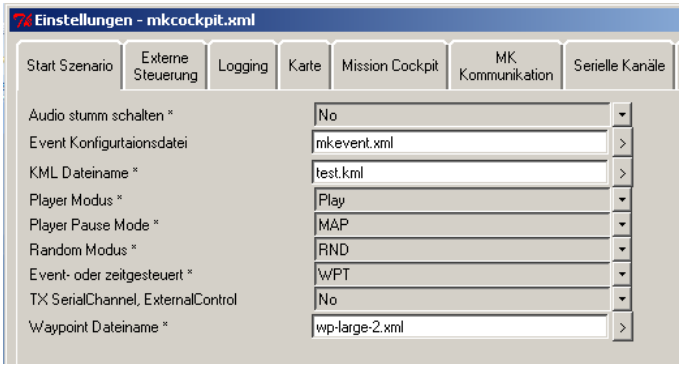
See **`mkcockpit.xml`** for possible Section and Key names.
The parameters can be used in the KML as well as in XML definition of maps.
Please note:
Be aware of upper/lower case for section and key names.

Attention:
Parameters that were read by the map definition can no longer be reprogrammed from the original settings of the **`mkcockpit.xml`**.
They are therefore also in the settings dialog displayed and stored when saved in the **`mkcockpit.xml`**. This overwrites the original parameters in the **`mkcockpit.xml`** with the data from the definition of the cards!

10. Start-Up Scenario

In the Preferences dialog, in the "**start scenario"** tab, you can specify default settings for the Mission Cockpit Player.

Thus immediately after the start of the program, the Player can be started, in a certain mode and an automatic waypoint and / or KML file to be loaded.



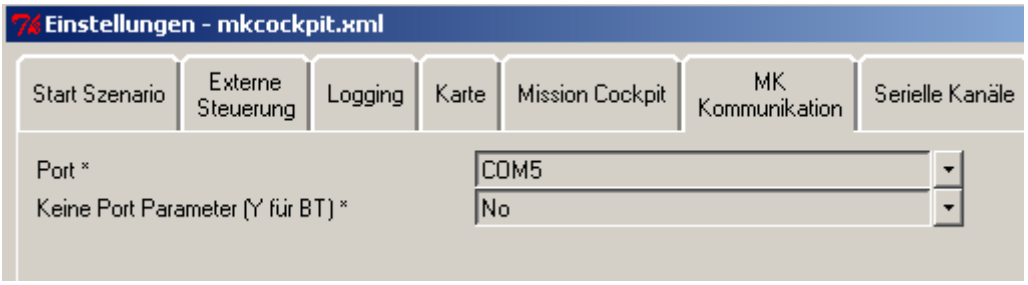
11. Data Link to the MK

For a reliable operation of Mission Cockpit **is** necessary a stable data-link connection to the debug port of the Navi-Ctrl**.** There are relatively transferred large amounts of data at a baud rate of 57.4 kbps**.** Communication takes place only with the Navi-Ctrl. The MK-OSD and MK debug data is subscribed with a frequency of 10 Hz. Thus a relatively fluid display is achieved on the screen. In Player mode, Mission Cockpit sends the target data sets with a frequency of 2 Hz. The serial channels and External-Control are sent with an adjustable frequency

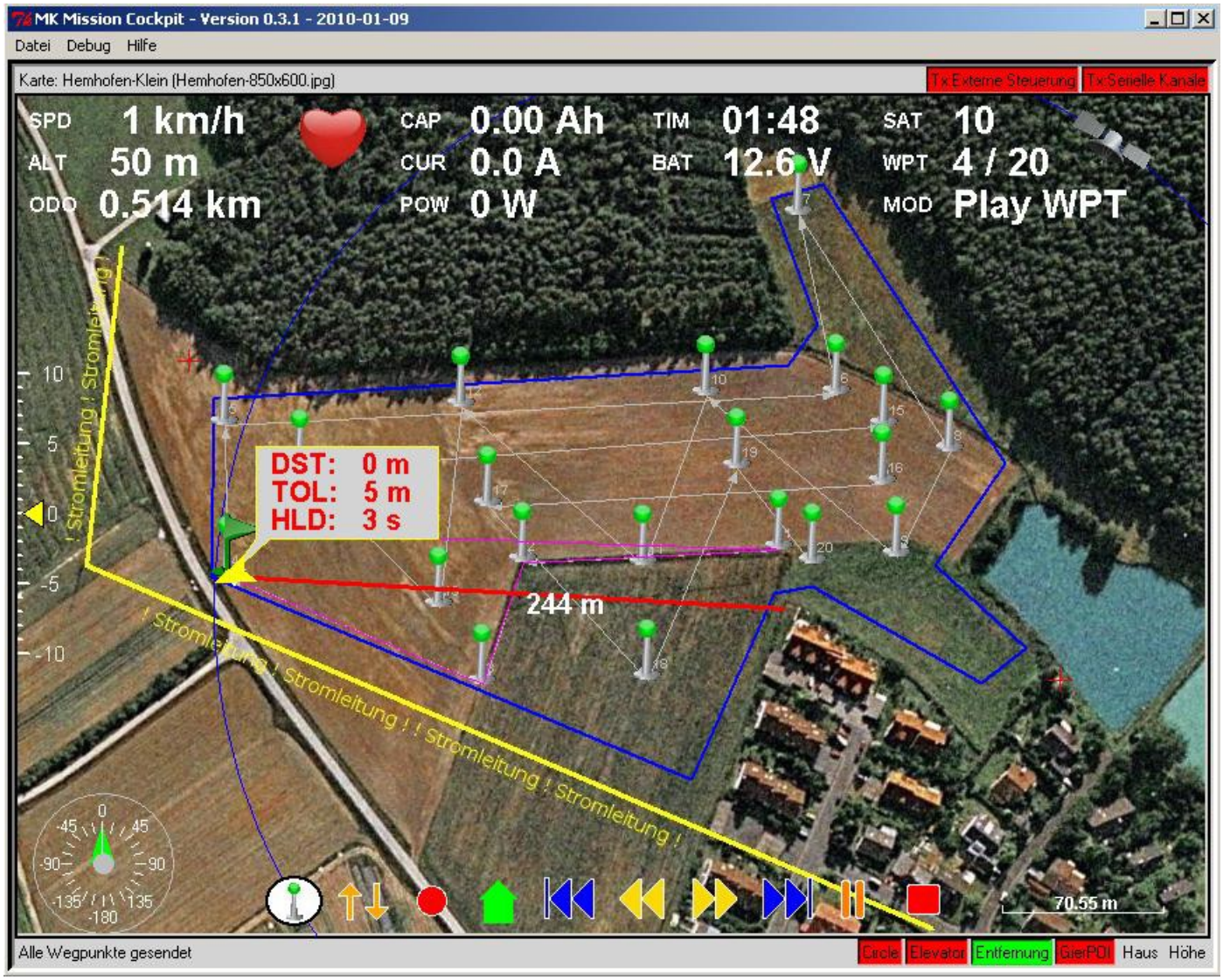
The WI.232 Module in 868 MHz band has been proved to be reliable, even at great distances (250 m).

Bluetooth is not particularly well suited because of range, as well as, from error-behavior (no Re-Connect at Disconnection).

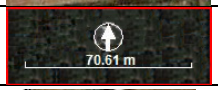
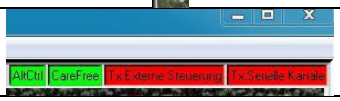
The COM port configuration for the data link take place in the preferences dialog in the tab "MK-Communication":

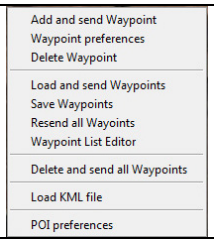


12. Objects shown on the Airfield Map



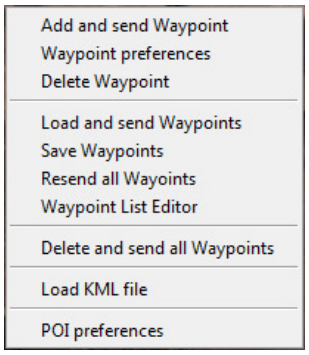
	OSD display of critical system data**.** The height amount in parentheses is the desired height specified number with Altitude Hold activated**.**
	Consumed Battery Capacity, Current and Power **Important**: Displayed values are based on data obtained from the BL-Ctrl. Deviations of 10% or more are common. The power consumption of the FC / NC and auxiliary facilities such as LED, wireless technology, FPV, servos cannot be measured. It is assumed to be a constant of 0.5A. This may be too little depending on the actual equipment of the MK. The capacity indication can be adjusted with the "Factor for Used Capacity" in the preferences dialog "Map". In my Quadro a correction factor of 1.07 (7%) had a very good agreement with the actual capacity consumption. **Warning:** **The indicated consumed Battery Capacity is thus not a reliable criterion for determining the actual remaining Flight Time.**
	Mouse-click over Flight Time or Odometer as to reset the values to 00: 00 and 0.0 km
	The MK is represented as an Arrow. The arrowhead shows the line of sight of MK The Arrow color changes depending on the number of received satellites: Red: No Satellite Orange: Less than 6 satellites

	Yellow: 6 or more satellites The white Arrow indicates the speed of the MK as direction and magnitude.
	"Rubber band" between MK to Home Position MK on Target with a distance indication
	Waypoint. The waypoint can be moved freely on the map by holding the left mouse button
	Target. When the MK has aimed at a target, then the destination point target is marked with the flag
	"Follow-Bear". The bear is displayed in the Player pause mode. You can move it with pressed left mouse button on the map. The MK then flies behind it.
	Position of the Tracking Antenna. Is only displayed if the Antenna Tracker is started.
	Point Of Interest (POI). In Player mode, the MK is aligned so that it always looks to the POI. The POI can be moved freely on the map with your mouse.
	Heartbeat The heart "beats" when the data-link to the MK is present and Data is received.
	Airfield Boundary. In Player mode Mission Cockpit will not fly to any targets, that will cross the border line in order to attain the Airfield Boundary limits **<u>Caution</u>:** With external influences, such as wind, it can still happen that the MK is driven off, leaving the airfield. **<u>Important</u>:** In SPD player mode or KML mode MK stops at the Airfield Boundary until the destination is reachable again.
	Important system messages are displayed as "Bubble Icons" directly to the MK Arrow.
	The Map scale is displayed at the right lower screen area
	Measuring function. With the left mouse button pressed, it shows a "rubber band". The length and absolute compass direction is displayed in the status bar.
	Buttons to control the mission cockpit player.
	If the tracking antenna is active, then left lower on the screen is displayed the viewing direction of the antenna.
	VSI - Vertical Speed Indicator Shows climb or descent of the MK
	Send status of "External-Control" and "Serial channels" in the top status bar: Green: Data is sent,

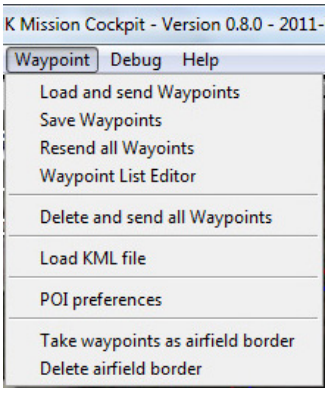
	Red: Sending is disabled in the Startup Script or RETURN key, Gray: Sending is permanently disabled, in Preferences Dialog. **Caution:** During takeoff and landing, sending over should be disabled by the RETURN key (= red), so that the program can not unexpectedly access the Flight Control System. "AltCtrl" and "CareFree" are Green, if the AH and/or CF control is turned on respectively.
	Status of the "Event Engine" events in the bottom status bar: Green: Event "Action" is active, Red: Event "Action / Else" is active, Gray: Neither "Action" or "Action / Else" is active
	Right-Click Menu

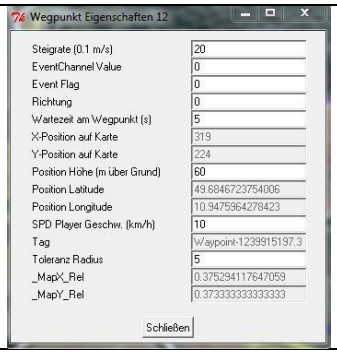
13. Waypoint Management.

Press the right mouse button (RMB) on screen, to access a menu with the following functions:



The functions, they are not directly related to a specific waypoint, can also be reached via the menu bar.



Right-Click Menu	Function
Add Waypoint and send	Set a waypoint at the current cursor position and transmits the waypoint immediately to the MK The waypoint can later be moved by drag / drop with the left mouse button. The waypoint connection lines are then displayed in red. This means that the displayed waypoints does not match anymore, with the already to the MK transferred waypoints. If Missions Cockpit Player is not being used, then, if necessary, the modified waypoints need to be retransmitted to MK.
Waypoint Properties	 Modifications are allowed to the white stored fields. The grey fields are managed only by the program itself.
Clearing of Waypoints	Deletes the indicated Waypoint under the mouse pointer.
Loading Waypoints and sending	Loads a File List of waypoints data and sends it to the MK. Can read Missions Cockpit XML Files and KopterTool WPL Waypoint Lists. When you load the waypoints stored in the waypoint file, relative x / y coordinates are used to determine the new positions on the map. The GPS coordinates are then recalculated with the calibration data of the current map and the relative x / y pixel coordinates. It is thus possible, to load a complicated waypoint model created on a different, larger or smaller, map.
Saving Waypoints	Saves the displayed waypoints in a XML format File. In addition to the GPS coordinates, also the x/y pixel coordinates are stored as floating point numbers relative to the size of the image in a range from 0.0 to 1.0.
Resending All Waypoints	Deletes previous waypoint list in the MK, and then sends a new complete path waypoint List.
Waypoints List Editor	Opens the waypoint Editor. (See below)
Deleting All Waypoints both from Screen & MK	Clears all waypoints displayed on screen and also erases the waypoint list in the MK memory.
Loading KML Track Data	Download a KML Track File and switch the Player into KML mode.
POI Properties	 Modifications are allowed to the white stored fields. The grey fields are managed only by the program itself.

13.1. Waypoint List Editor.

With the **"Waypoint List Editor"** can the properties of individual waypoints, or all of them simultaneously, modified and updated.

WP Nr	Heading	Altitude	AltRate	Speed	Wait	Toleranc	Ev.Ch	Ev.Flag
Set All	---	---	---	---	---	---	---	---
1	0	5	20	10	10	5	0	1
2	0	10	20	10	5	5	0	0
3	0	15	20	10	5	5	0	0
4	0	20	20	10	5	5	0	0
5	0	25	20	10	5	5	0	0
6	0	30	20	10	5	5	0	0

You can change the properties of a waypoint using the **"Waypoint List Editor"** and/or by pressing the right mouse button to open a dialog for a specific waypoint.

❖ **The two methods are synchronized.**

Here an input option, to enter GPS coordinates directly, does not exist.
The GPS coordinates are set only by moving the waypoint icons on a map.
Loading and saving of waypoints is take place in the menu "Waypoints" or with Right-Click Button menu.

The **"Waypoint List Editor"** offers the following **features**:

- ✓ Changing a single Waypoint Cell
- ✓ Copy / paste individual cells (Ctrl-C Ctrl-V) Copy / paste multiple cells (Ctrl-C Ctrl-V).
- ✓ Change all cells in a column by typing the first line (set all).
- ✓ Entering formulas (Perl program) in the first row (set all).
- ✓ You have access to all Mission Cockpit / Perl functions and variables.
- ✓ The variable **`$Wp`** is the current waypoint number.

Example:

Input	Results
`123`	Sets all fields in the column to the value "123"
`$Wp`	1, 2, 3, 4, 5, ...
`$Wp * 2 + 10`	12, 14, 16, 18, 20, ...
`int rand 30`	Calculated for each cell of the column as a random number between 0 and 30 This can for example specify random altitude or heading-values.

14. Waypoint Fly – Classic Navi Ctrl mode.

In this mode, the flying of the waypoints is executed according to the Navi-Ctrl rules.
Available Tasks on this mode:

- ✓ Managing Waypoints
- ✓ Transfer of waypoints from/to MikroKopter
- ✓ Visualization of the tactical Flight Data

The MikroKopter flies the waypoints automatically, if **WP/CH mode** is switched on by remote control.
The Navi-Ctrl can currently handle max. 20 waypoints records (from NC 0:18 there are max. 30 records).
Being created in the Mission's Cockpit screen more waypoints, then only the first 20/30 waypoints are considered by the Navi-Ctrl.

15. Waypoint Player – Controlled from Mission Cockpit.

The implemented in Mission Cockpit **Waypoint Player** offers much more possibilities than the classic one, controlled by the NC Waypoint Flying Engine.

The **Waypoint Player** does not transfer static waypoint lists on the MK, but sends two target records per second, with the next destination to the MK.
Thus, the player can respond quickly and very flexibly to the requirements of the pilot.

This will require **a very stable data link** to the MK!
If the data link **fails**, then the MK waits a 60 sec Hold Time, and after returns automatically, to the home position.

To fly with the **Waypoint Player** the MK needs to be switched to **"Coming Home mode",** (CH), via a remote control, as to be flying a classic Waypoint File.

Because the **Waypoint Player** constantly transmits the position of the next destination point to the MK, the Waypoint List constantly changes as new waypoints are added / modified while the MK already executes the list, or even changing the in-progress played waypoint.

Therefore a re-transmission of the waypoints, as in the classic MK, is no more required. The waypoint player uses the same waypoint lists like in classic waypoint flying.

- ❖ **There is however, no limit to the number of waypoints.**

The handling of the waypoints using the Right-Click menu is the same as for the Classic Waypoint flying and is described in chapter 13/14 above.

(It is annoying that the MK when a new target point is set, beeps short. Since the MC transfers records twice per second then it is very annoying. That is why I have muted the beeper with a piece of adhesive tape. Actually you do not need this, because the relevant data are displayed or announced through Mission's Cockpit speech engine.)

The Player can be operated in three different modes:

15.1. Event driven Waypoint Player.

The MK flies straight to the target Waypoint. When he reached the WP (WP-tolerance), he stops for a while (hold time) and then flies on to the next waypoint.

There are the following options:

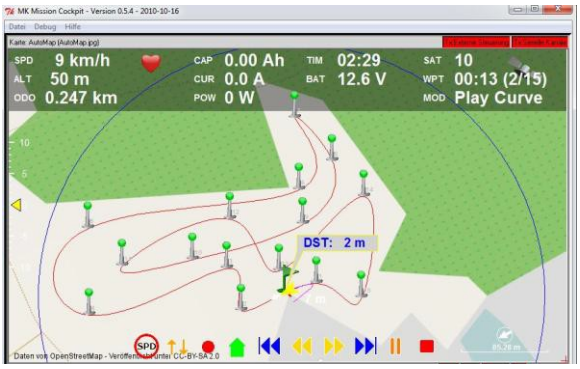
- **Play WPT** - The waypoints must be flown in sequence.
- **Edge WPT** - The waypoints are approached randomly. The lines connecting waypoints are hidden.
- **Edge MAP** - Are randomly approached arbitrary coordinates on the map. The 10% of the card-edges is generally not used. The Waypoint symbols and connecting lines are hidden. If there are any obstacles should be placed on the map as to be shown on the airfield. The airfield boundary can be configured.

15.2. Speed controlled SPD Player.

SPD stands for "Speed".

- The MK flies to the waypoints with the specified speed on Waypoint properties.
- The speed control is done by dynamic setting of target points at a short distance in front of the MK.
- The player waits for the MK, if the distance from the MK to the destination point is longer than a distance that would be covered in 6s flight time.
- When the flight path cross the Airport Boundary, MK remains at the Boundary until the trajectory is again inside the airfield and next waypoint is reachable.

The waypoint trajectories are not like the WPT mode straight lines, but connected with a spline curve (more precisely a "closed natural cubic spline"). This allows a "smooth" fly off the flight path, without having to slow down on the waypoints.



15.3. Time controlled KML Player.

- The MK flies a track with a fixed time basis.
- The track can be downloaded either from a KML file or originates from the Mission cockpit recording function.
- Normally a KML does not include any time base information. The points of the KML tracks are reached with a time base that can be set by "waypoints" configuration **KML player time base**.
- When the flight path cross the Airport Boundary, MK remains at the Boundary until the trajectory is again inside the airfield and next waypoint is reachable.

The waypoints are hidden and instead the loaded KML track is displayed.

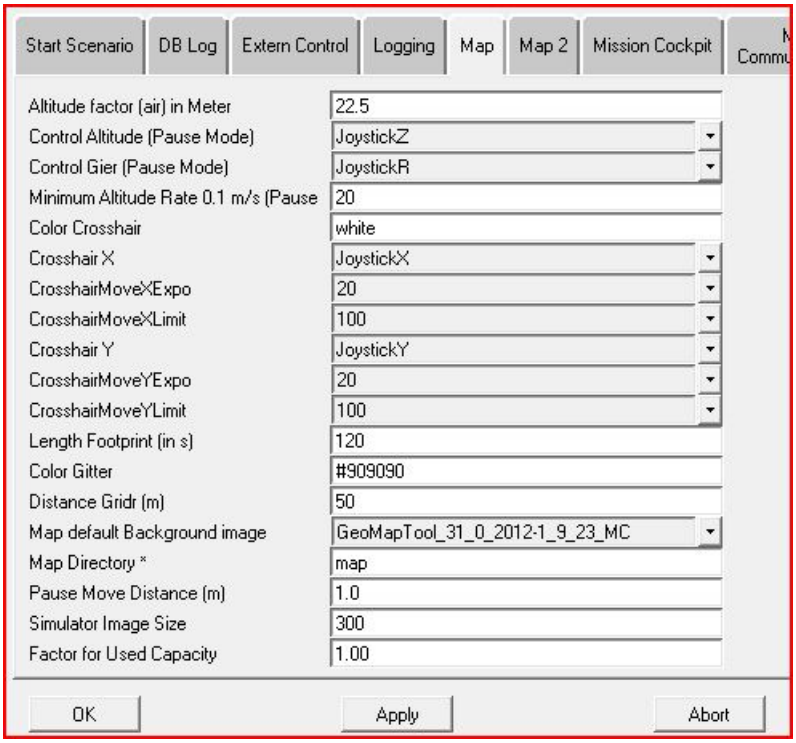
15.4. Pause Mode.

The Pause mode can be activated in all player modes (Space Bar). When activated the UAV remains at the current position (Position Hold).

With the cursor keys, joystick or 3D mouse, the MK position can be changed based on the map or MK-coordinate system. Thus, a precise- GPS based positioning is possible.

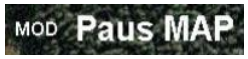
The "Follow-Bear" icon is displayed and can be moved with the mouse. Then MK flies following it.

In Pause Mode the following control options can be used:

<table>
<tr><th>Control Type</th><th>Description</th></tr>
<tr><td>Positioning by Keyboard keystrokes</td><td>The position can be changed with 1 meter per keystroke using the cursor keys.
The move distance per keystroke can be set in:
Preferences dialog→ Tab "Map" →"Pause Move Distance (m).</td></tr>
<tr><td rowspan="3">Positioning via Joystick or 3D-Mouse</td><td>In the Preferences dialog→ Tab "Map", you can enable the input device for positioning a crosshair on the map using: "Crosshair X" and "Crosshair Y".</td></tr>
<tr><td> 
</td></tr>
<tr><td>Also here can be set an Expo value and Travel Limit.</td></tr>
<tr><td>Altitude input via keyboard</td><td>With the keys PageUp / PageDown the desired Altitude can be changed 1m per keystroke.</td></tr>
<tr><td>Altitude input via Joystick or 3D-Mouse</td><td>In the Preferences dialog→ Tab "Map", you can enable the input device in "Control Altitude (Pause Mode)".</td></tr>
<tr><td>Direction input via keyboard</td><td>With the <> the desired direction can be changed by 5 degrees per keystroke.</td></tr>
<tr><td>Direction input via Joystick or 3D-Mouse</td><td>In the Preferences dialog→ Tab "Map", you can enable the input device in "Control Gier (Pause Mode)".</td></tr>
</table>

The crosshair will appear automatically as long as a control function is enabled from the keyboard or Joystick/3D-Maus. After 5 sec, if a control activity has not been detected, the crosshairs will disappear automatically. In the meantime, the MK have reached the submitted goal and targets to the target icon that will displayed on the map.

The position control can be made with respect to different coordinate systems. Switching is done by pressing the "x":

Based on the pixel coordinate system of the illustrated map.	
Based on the current viewing direction of the MikroKopter.	

The default direction is indicated by a dashed line outgoing from the center crosshair

15.5. Waypoint Player Control.

The control of the waypoint player is basically just like a CD player. The operation is done either via:

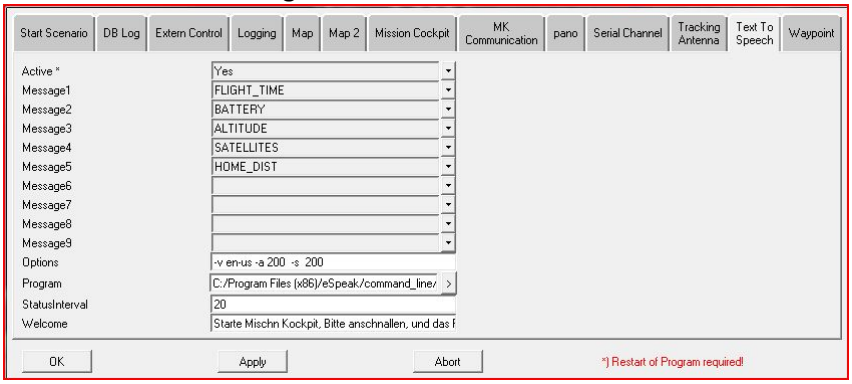
- Mouse and the on-screen icons at the bottom.
- Keyboard (should be easier to use it while flying with a mouse).
- A programmable remote PC that simulates keystrokes.

  I'm using for example an X10 wireless remote control that often accompanies Medion PCs to control Media Center, or in connection with PS3 REMOTE CONTROL EventGhost (www.eventghost.org).

  This is my preferred method.

The Microsoft Vista/Win7 voice control can simulate keystrokes.

Button/Symbol	Key	Function
	Spacebar	Switches between **Start** and **Pause** mode. In pause mode MK remains to its current position.
	S	Stops the Waypoint Player and returns to the classic Navi-Ctrl mode.
	N	Continue to next waypoint. In the KML mode forward 10s. In the SPD mode, forwards the flight time by 10s.
	P	Back to the previous waypoint. In the KML mode 10s Rewind. In the SPD mode, backwards the flight time by 10s.
	L	Next to the last waypoint.
	F	Back to the first waypoint.
	0 - 9	Fly to the waypoint with the number you entered on the keyboard. For two-digit numbers, the number is entered as normal. The number is accepted, when within 0.7s time period no other key is pressed.
	H	Fly to the home position. The home position can be either: ▪ The Start Position. ▪ The in the map definition given Home Position.
	A	Toggles the recording of the flight path **ON** or **OFF**. The track can then be played in KML mode. The recording can be made in any flight modes, including on free mode. The recording may override a loaded track of KML file. On new activation the old record will be overwritten. It is not intended to save a recorded track. For this use the logging module.

			R	When on the WPT mode, switches between Random modes: ✓ **Play WPT** The waypoints must be flown in sequence. ✓ **Edge WPT** The waypoints are approached randomly. ✓ **MAP Edge** Are randomly used any coordinates on the map. **Attention**: Applies only in the WPT mode, not in the SPD or KML mode.
			w: Forwards k: Backwards	Switches the player mode to choose between: ➢ **WPT**: Event-controlled. ➢ **KML**: Time Scheduled. ➢ **SPD**: Speed Controlled.
None			m	Turns the TTS voice output On / Off. If the speech is to be used, the program espeak must be installed. http://espeak.sourceforge.net There are regular announcements of Flight Time, Voltage, Altitude, number of received Satellites, etc. The order, content and time spacing of the announcement can be adjusted in the Preferences dialog:  In the message lines can also be entered Perl statements, for example to calculate the message text or format. Fixed text (strings) must be bracketed with quotation marks, e.g. "End of story." Important system messages, for example Battery Warning, leaving the Airfield, Reception Problems are prioritized so cannot be configured.
None			v	Switches the **POI** mode On / Off. In the POI mode, the viewing direction of the MK always points to the POI.
None			g	Toggles a **Grid** on the map On / Off. The spacing of the grid lines can be set in the Preferences dialog.
None			Return	Disables / Enables sending of "Serial Channels" and "External Control". **Important**: During takeoff or landing should sending be disabled, so the program cannot interfere with the Flight Controls.
None			C	Sets the current position of MK as a waypoint.
None			ESC	Terminates Mission Cockpit.

15.6. Yaw and Altitude Control.

Since version 0.7.0, the yaw and altitude control is implemented in Missions Cockpit core.

Important:
The yaw and altitude control is only active when the MK is in Carefree mode. In an emergency, you can turn off the yaw and pitch control, by switching off the Carefree on the RC transmitter.

The Altitude system control behaves as follows depending on the player mode:

Altitude control behavior	Play WPT Edge WPT	Edge MAP	KML	SPD	Pause	Home
Altitude control available	✓	✓	✓	✓	✓	
Altitude control on Waypoint	✓			✓		
Altitude control through KML data or Missions Cockpit recording			✓			
Taking actual Height as Target Height on mode activation		✓			✓	
The Target Height is set only when the first sighting of the target WP	✓					
The current target level is continually recalculated and adjusted			✓	✓	✓	
RC manual override is possible	✓	✓				✓
Manual override with PageUp / PageDown, joystick, or 3D-Mouse	✓	✓			✓	
Using the climb rate of Target WP	✓			✓		
Using the climb rate from the configuration		Default WP climb rate	KML climb rate			
The climb rate is dynamically determined					✓	

When Altitude system control is active then the Target Height is indicated in OSD after the actual height in Parentheses 

The climb rate is indicated in 0.1 m/s (e.g.: 20 = 2 m/s)

The Altitude control is disabled if the climb rate is "0".

In Pause mode, the climb rate is dynamically calculated from the height difference between actual and target height.

- Maximum Climb Rate: 60 (= 6 m/s).
- Minimum Climb Rate: from configuration.

In SPD mode, the system continuously calculates the elevation data from the start and finish point, to determine the target altitude for the current position of the MK. Thus an elevation profile can be flown relatively accurately.

The Yaw system control behaves as follows depending on the player mode:

Yaw control behavior	Play WPT Edge WPT	Edge MAP	KML	SPD	Pause	Home
Yaw control available	✓	✓	✓	✓	✓	✓
Yaw control on Waypoint	✓			✓		
Yaw control through KML data						
Yaw control through Missions Cockpit recording			✓			
Orientation to the POI	✓	✓	✓	✓	✓	✓
RC manual override is possible						
Manual override with </> Keys, joystick, or 3D-Mouse	✓	✓	✓		✓	

The orientation of the POI has the highest priority. The POI mode is activated/deactivated with the V key.

16. Antenna Tracking.

In Mission Cockpit is implemented control for Pan/Tilt Antenna Tracking. Thus, for example, a directional antenna or a video tracking path can be driven.

Following Items will be required:

- **Pololu Micro Serial Servo Controller** http://www.pololu.com/catalog/product/207
  Available e.g. from:
  http://www.shop.robotikhardware.de/shop/catalog/product_info.php?cPath=65&products_id=118
  http://www.nodna.com/product_info.php?products_id=721&XTCsid=ljn96a9crlqk37cv2okv27qgo2

Or:

- **Pololu Micro Maestro 6-Channel USB Controller** http://www.pololu.com/catalog/product/1350
- **Two servos for Pan and Tilt, which allow a 180 degree mechanical angle rotation.**
  The servo can be controlled outside the usual range of about 0.25-2.75ms **to reach the 180 degree angle of rotation. It can be used quiet cheap, slow servos, with the appropriate mechanical force.**
- **One appropriate Pan/Tilt mechanism.**

16.1. Pololu Micro Serial Servo Controller.

The power board is connected to a serial port on the PC.

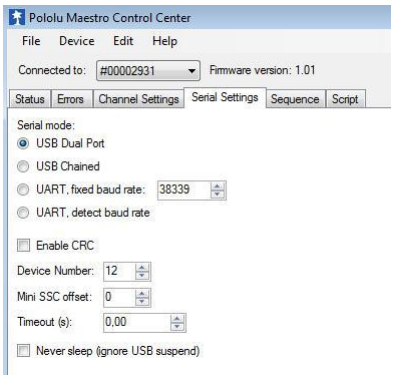
Pin assignments for the Sub-D 9-pin RS-232 connector:

```
Pin 3   ->   "RS-232 serial input" from the servo board.
Pin 5   ->   "GND" from the servo board.
```

The servo board is controlled in Pololu mode; also the jumper is not connected.

The Pan Servo (horizontal) is connected to the first Servo connector.
The Tilt Servo (vertical) is connected to the second Servo connector.

16.2. Pololu Micro Maestro 6-Channel USB Controller.



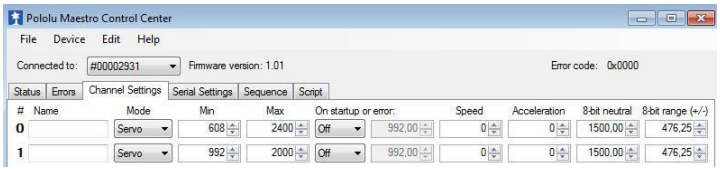
In the Pololu Control Center one of the two USB modes must be set, so that the controller accepts commands from the USB port.

In the Mission Cockpit must be set the virtual COM port from the Pololu command port.

In the Pololu Control Center must be set for the used servos 0 and 1 the maximum servo pulse width.

In my setup were approximately 600-2400µs; with 1500µs at the center position.

The pulse width must be greater than or equal to the Missions cockpit configuration.



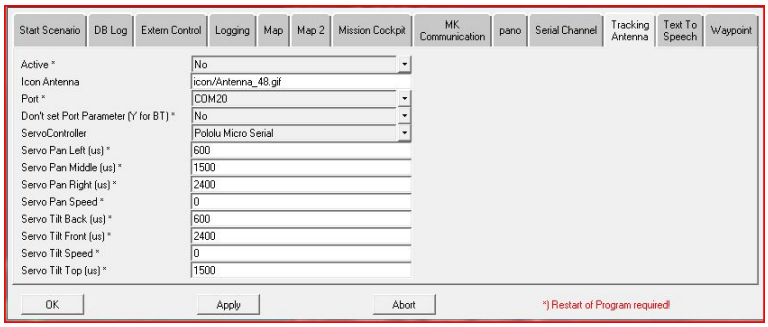
The Pololu control center can be run in parallel to the Mission Cockpit and observe the movement of the servos in the Tab "Status".

16.3. Configuration.

Info: As of version 0.4.2 the parameters have changed.

The tracking antenna is configured and activated (Y / N) in Preferences dialog, tab "Tracking Antenna".

Here also are set:
the COM port and the Servo Parameters



`Servo Controller.`	Selection of the servo controller: "Micoro Pololu Serial" or "Pololu Maestro"
`Don't set Port Parameters (Y for BT).`	When activated, the Baud rate is not set to the COM port. Needed for example for a Bluetooth virtual COM port.
`Servo Pan Left/Middle/Right,` `Servo Tilt Front/Top/Back.`	Here are set the servo pulse widths to adjust the servo-center position and servo-end positions. The servo end positions must be mechanically ±90 degrees from the middle position. Thus the servo can rotate 180°. The servo-center position (center / top) typically is at 1500µs. For the standard pulse width of 1000-2000µs (i.e. ±500µs, starting from the middle position) results depending on the servo-type a rotation angle of approximately ±45°-55°. If the servo rotation must be reversed, then the fields Left/Right and Back/Front must also be reversed. The pulse widths for the servo limit positions must be determined by experiment. One can start i.e. with ±900µs those results to 600-1500-2400. The servos **must not reach** the mechanical hard limits. **Important:** Modified settings are not take effect until you restart the program.
`Servo Pan Speed,` `Servo Tilt Speed.`	0 = Max. Speed. 1-127: Changing the pulse width in 0.5µs increments. i.e.: 1: Slow 127: Fast

If the tracking antenna is activated, a direction compass appears on the left lower corner on the map that simulates the antenna direction.

The Tracker calculates the required antenna orientation, from the current antenna location plus the GPS position and altitude received data from the MikroKopter's Data Link.

Thus the Tracker, as to aim the Antenna, requires the GPS coordinates, GPS altitude and Compass direction from antenna's center position. The Tracker acquires the required data from MikroKopter. For this the MK must be placed in front of the antenna, aiming exactly in the same direction as the viewing direction of the antenna in the middle position. At the start of the MK-engines, the data are taken from the MK.

If the location of the antenna from the starting point of MK differs, these values can be stored also optionally in the map definition, see Section 7.2.

The Antenna can also see "backwards". The Pan and Tilt-servos "flip" it at 180 degrees. This should be considered in the mechanical design of the Tracker mechanism.

16.4. Standalone Operation without User Interface.

The Tracker can also run standalone without the Mission Cockpit interface directly from the Cmd Shell.

Program call **`track.pl <optional Parameter>`**

Command line parameters	Description
`-TrackPort COMn`	COM-Port of Servo-Board.
`-MkPort COMn`	COM-Port of MK Data-Link.
`-ServoPan Left,Mid,Right`	Servo pulse widths for Left Middle & Right.
`-ServoTilt Front,Top,Back`	Servo pulse widths for Front Top & Back.
`-PanSpeed` `-TiltSpeed`	0 = Max. Speed. 1-127: Changing the pulse width in 0.5µs increments. i.e.: 1: Slow 127: Fast

If not additional parameters are specified, then the data is taken from the MikroKopter.
The following message at the start, can safely be ignored:

`Map "" not found in map.pl. Using "Default" map`

To test the antenna, the direct call can be very useful, because immediately after program start, the tracker executes the servo test sequence and the servos start moving.

17. Simulator.

Since version 0.2.7 Mission Cockpit contains a MikroKopter simulator. So you can try out the function and behavior of the program without compromising the MikroKopter in flight. The simulator is also very good for testing and adjusting the antenna tracker mechanism.

The simulator is started in the Mission Cockpit Debug menu. It runs in a separate top-level window and is almost independent of the main program.

The simulator writes directly to the internal OSD record hash. This is the same data that is sent in the normal operations of the MK. Furthermore, the simulator reads the mission from the cockpit calculated target records that are also sent to the MK.

At simulator mode the MK should be off, otherwise, there is data conflict in the OSD dataset.

The COM interfaces should be active because Mission Cockpit even in simulator mode sends data to the COM interfaces.

The controls shown in the Simulator correspond to the data elements sent from the MK to the OSD datasets.

The simulator is thus basically a graphical editor for the OSD data.

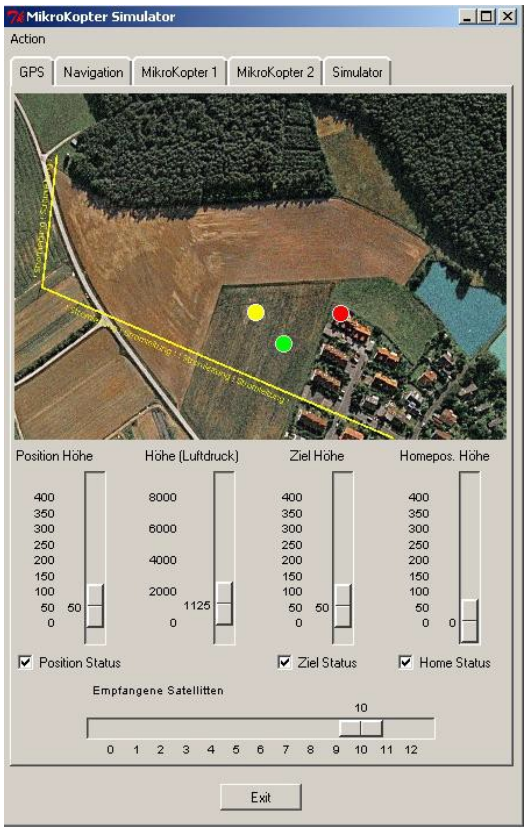
There are only the evaluated data of the OSD dataset, from the Mission Cockpit, displayed.

Because the input of GPS coordinates with sliders or input boxes are not particularly easy to use, the simulator has a scaled down visualization of the Airfield Map.

On the map the positions of the objects are represented as colored circles:

- MikroKopter: **Yellow**
- Target: **Green**
- Home Position: **Red**

The circles can be moved with the mouse and thus the GPS positions adjusted. With a little skill, the positioning is very good.



17.1. Simulator Manual Mode.

After starting, the Simulator is in manual mode. Now, the user has the task to simulate the OSD data usually sent by the MK. Response of different inputs is represented live on the map of Mission Cockpit. You can also set conditions that normally do not occur.

Some settings need to be made as to bring the MikroKopter to "Fly". This requires knowledge of the functions of MK and the OSD dataset. To facilitate this, there are some macros that make this work for you on the action menu.

How to start could e.g... look like this:

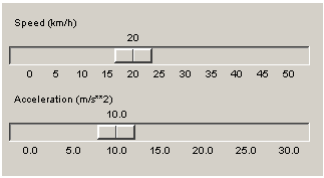
- ✓ **Action** → **3D Fix** activates satellite reception.
- ✓ On the Simulator Map place the Yellow MK icon to the desired location as the Home Position.
- ✓ Optional, for tracking antennas user, set the compass direction in the tab "Mikrokopter".
- ✓ Again **Action** → **3D Fix** resets the Home Position to the current position of the MK.
- ✓ **Action** → **Make MK Fly** starts the Gyro calibration sequence, starts the motors … the MK starts flying.

17.2. Simulator Automatic Mode.

After activating the simulator in action menu, the simulator takes flight control. He reads the transmitted target data from the Mission Cockpit and calculates the OSD dataset.

Important:
In simulator mode, the behavior of the MK is simulated as if Mission Cockpit it is in player mode. Therefore, Mission Cockpit must be switched in player mode. The classic Navi-Ctrl mode is not simulated.

The flight model is very simple. It simulates a motion with a constant acceleration and maximum speed, with deceleration on target.



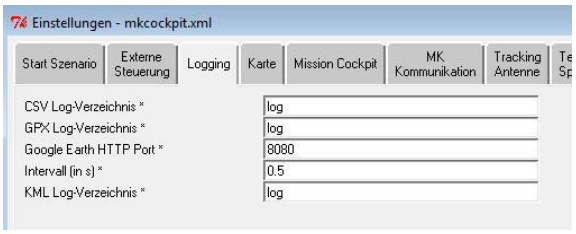
The Max. Velocity and Acceleration can be set in the **"Simulator" Tab**.

18. Logging.

Mission Cockpit automatically logs the following data locally on the PC:

- All data of the NC-OSD and NC-Debug datasets in CSV format.
- Track in KML format.
- Track in GPX format.

In the Preferences dialog, can be set the Storage directory and the logging interval.

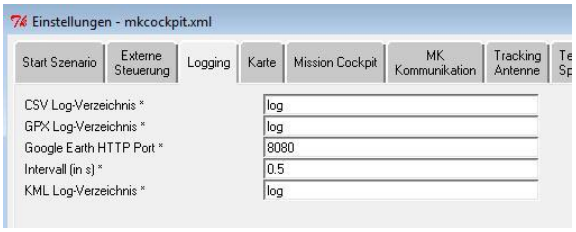


The interval can be as low as 0.1s, sometimes very useful to track short-term errors, e.g. at temporary reception failures.

If the MK is "lost" by accident, then the server-based logging with the GPS positions can be very useful for finding the MikroKopter. In simulator mode, the logging is disabled.

19. Google-Earth Server.

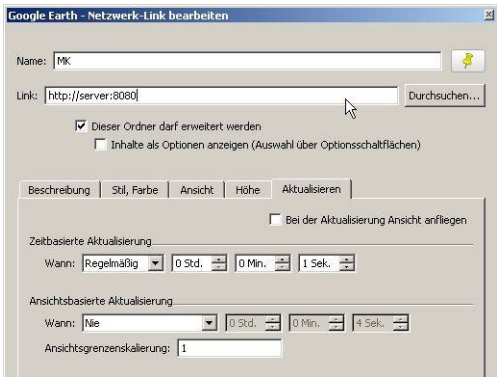
Mission Cockpit has a built-in Web server that can provide the KML-Tracks for Google Earth client. Thus, the flight can be observed over the Internet.



The IP port of the web server must be set in the Preferences dialog of Mission Cockpit.

In Google Earth client a network link must be created.

In the Update tab set an update interval of 1 sec.

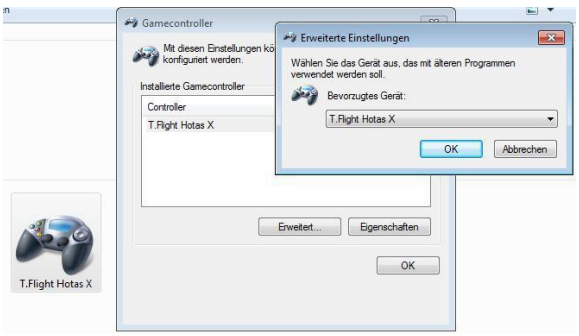


20. Input Devices & External Control.

Since version 0.4.0 Mission Cockpit supports joystick and / or a 3D-Mouse as input device. The control of the MikroKopter is carried out via the "serial channels," or the "External Control" software interface on the NC.
The input devices can also be used to control the crosshairs in the Player Pause Mode (Position, Altitude, Direction).

20.1. Joystick as Input Device.

Standard PC-Joysticks with max. 6 analog axes and 12 buttons/switches are supported.



Important:
It uses the joystick, which is set as the preferred device in the Control Panel, Advanced Game Controller settings.

The current joystick data can be observed in the debug menu "Joystick/3D-Mouse".

20.2. 3D-Mouse as Input Device.

The 3D-Mouse "Space Navigator" from 3Dconnexion is supported. The mouse has 6 analog axes and two Control Buttons.

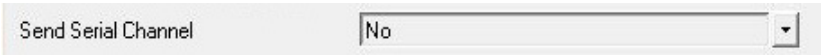
The current 3D-Mouse data can be observed in the debug menu "Joystick/3D-Mouse".

Actually, one should not recommend this mouse because the manufacturer delivers it without a standard joystick or mouse driver. The application must directly support this device. These are typically only CAD programs and Google Earth. Otherwise, you can use it only as "chic paperweight". It has cost me a lot of nerves, to link this mouse over an EXE-Program under the directory "bin" with Mission Cockpit.

20.3. Control over "Serial Channels".

Since FC 0.78 / NC 0.18, MikroKopter has 12 **"Serial Channels"** that can be controlled by the Mission Cockpit. Then, in the MikroKopter Tool, you can associate the appropriate serial channels to a control function.

If the serial channels should be used, then "Send Serial Channel" must be enabled in the Preferences dialog:



The transmission status is shown in the top status line:

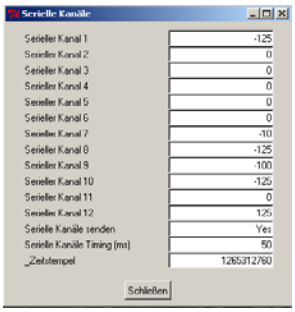
Not Sending
Not Sending
Sending

In addition, with the **RETURN** Key, can be switching between **"Not Sending"** and the Service Mode set in the Preferences dialog.

Important:
During takeoff and landing, sending over should be disabled by the RETURN key (= red), so that the program can not unexpectedly access the Flight Control system.

The MikroKopter configuration does not allow you to use a serial port to control the main flight controls (Roll, Pitch, Yaw and Throttle). However, this can take place via the "External Control" function ... see the next chapter.

The data sent to MikroKopter can be observed in the debug menu "Serial Channels".



The range of serial channels goes from -125—0—125

20.4. Control over the"External Control".

Since NC 0.18 the navigation via **"External Control"** over the NaviCtrl is possible. Thus Mission Cockpit can directly access this control functions (Roll, Pitch, Yaw, Throttle and Altitude) and take over.

If the external control should be used, then "Send Extern Control" must be enabled in the Preferences dialog:

The transmission status is shown in the top status line:

Not Sending
Not Sending
Sending

In addition, with the **RETURN** Key, can be switching between **"Not Sending"** and the Service Mode set in the Preferences dialog.

Important:
To activate "External Control", a value greater than 128 must be set to the "External control" in MikroKopter Tool.

⚠ CAUTION

Activating the "External Control", the Mission Cockpit program takes direct control of the MK flight control functions. There may be hazardous situations to which the Users must respond accordingly!
Beginners should not enable "External Control"! This is only for users who know what they are doing and are aware of the risks!

⚠ CAUTION

During takeoff and landing, sending over, should be disabled by the RETURN key (= red), so that the program can not unexpectedly access the Flight Control system.

⚠ CAUTION

The control of Roll / Pitch / Yaw / Throttle through "External Control" and the Serial Data Link is considered experimental. For direct control of MK, the data transfer is too uncertain and too slow.

⚠ CAUTION

"External Control" should only be enabled if the user is familiar with the functions and the behavior of the system.

⚠ CAUTION

Function of "External Control" must be disabled in the Mission Cockpit Preferences dialog, and preferably in the MikroKopter Tool when not using this functionality.

⚠ CAUTION

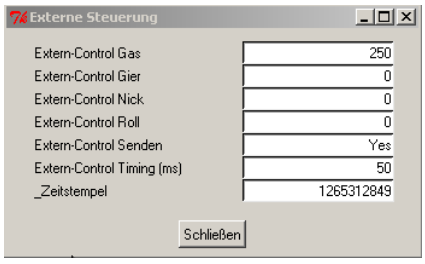
Note the following considerations when "External Control" is enabled:

- The throttle stick on the transmitter determines the maximum possible acceleration. The throttle stick must be so correspondingly high, so that the Mission Cockpit can give also more acceleration.
- In reverse, the RC throttle stick cannot give more, than is dictated by Mission Cockpit using External Control.
- Pitch/Roll/Yaw act "additive" to the RC sticks.

For the Channels that you do not want to be controlled by the program, you should indicate them as follows:

Channel	Assignment in the Settings menu
Pitch	0
Roll	0
Yaw	0
Throttle	255

The current data sent to MikroKopter can be observed in the debug menu "External Control".



Range of values:

Pitch / Roll / Yaw: -125—0—125

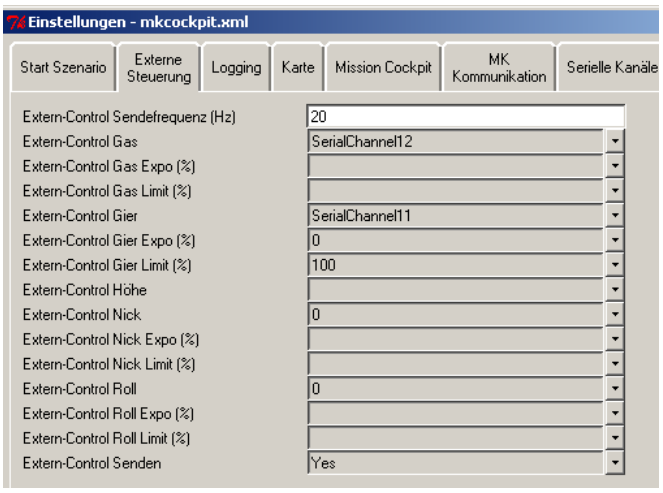
Throttle: 0—250

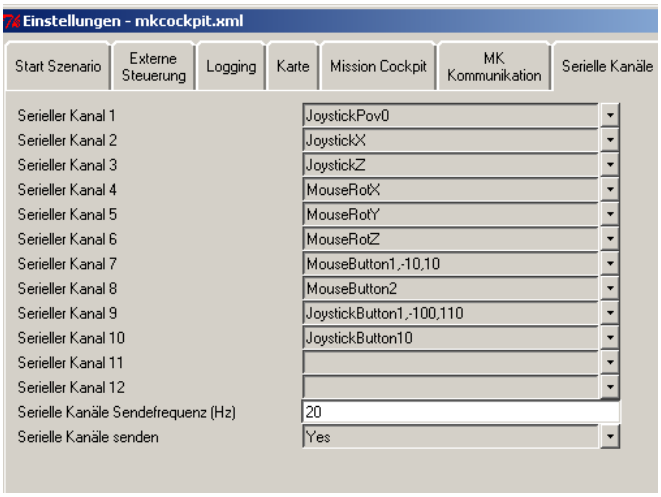
20.5. Configuration "Serial Channels", "External Control" and GPS Navigation.

The output channel mapping of the input device (joystick, 3D mouse, etc.) used for GPS navigation can be entered in the Preferences menu, of "External Control" and "Serial channel", as well as in the Preferences menu "Map" for Crosshair X , Crosshair Y, Direction and Altitude.

The output channels have a symmetric range of --125—0—125, with 0 = Center Position. Only the output channels, "External Control: Gas" and "External Control: Height" have an asymmetric value with range from 0—250.

The following used percentages of the parameters refer to the respective range of values of the input or output channels





In the simplest case, the most common input channels can be selected in the Options menu of the Preferences dialog:

Setting	Description

Analog	Available axes, Range (-125—0—125):
	`JoystickX/Y/Z/R/U/V`
	`MouseRotX/Y/Z`
	`MouseTranX/Y/Z`
	`SerialChannel01–12`
	`RcPoti1–8`

Analog, Reverse Direction	Same as Analog but with Suffix: `_Reverse`

Switching Function	Available switches, Value -125=OFF 125=ON
	`JoystickButton1..12`
	`JoystickPov0/45/90/135/180/225/270/315`
	`MouseButton1..2`
	`FctKey1..12`
	`FctKey13..24` (Shift F1..F12)
	`FctKeyToggle1..12`
	`FctKeyToggle13..24` (Shift F1..F12)

Fixed Value	i.e.: "-50"
Serial Chanel TX Frequency (Hz) External Control TX Frequency (Hz)	Here the frequency is set, with which the datasets are sent to the MK. The frequency should be not too high to avoid flooding the data link.
Serial Chanel Send External Control Send	**Important**: This option must be enabled if you want sent the data to the MK
RC Chanel Request	If RcPoti1—8 should be used, must this option in the tab "MK Communication" be activated, so that the pots are requested by the MK. **Important**: The query of the pots charged the data downlink with approximately 320 Bytes/s and should only be enabled if the pots are really needed.
___Expo% ___Limit%	"External Control" and the "Serial Channels" can be configured with an Expo and travel limit. The parameters in the Preferences dialog for reasons of space are disabled on serial channels. You can however re enabled them by supplementing of "SerialChannelNNExpo" and "SerialChannelNNLimit" in the mkcockpit.xml.

Mission Cockpit offers also further configuration options that affect the input channels or the respective output channel. It can also combine multiple input channels and the sum of the output channel to be evaluated.
The configuration must in this case by manual entry carried out in the respective input field.

Parameter Syntax of the input channels (analog and switch). The parameters are separated with commas:

`Control_reverse, Min%, Max%, Expo%, Offset%`

Input Channel Parameter	Description
Control	Name of the input device, see above, e.g. JoxstickX
_reverse	Optional, if the effective direction of the input channel must be inverted
Min%	Linear travel limit in % of Stick in negative direction. **Note:** The parameter is indicated as a positive number although thereby the negative path limit is set. With a negative value an inversion is possible.
Max%	Linear travel limit in % of Stick in positive direction. With a negative value an inversion is possible
Expo%	Evaluation of the input channel with an exponential function. Parameter in%: -100—0—100 (0% = neutral).
Offset%	Offset in% **Note:** The RC-potentiometers RcPoti1—8 are different from the other analog input channels, having an asymmetric range of values of 0—235, with the center position in 110. With an offset of -88% can be made a Poti symmetrical.

The functions are evaluated to the input channels in the following order:

1. Reverse.
2. Expo.
3. Linear Travel Limit Min/Max.
4. Offset.

Parameter Syntax of the output channels. The parameters are separated with commas:

Property, Par1, Par2, Par3, Par4, ... ParN

Property / Parameter	Description
Inc	The incrementally mode is enabled for the output channel. The output channel follows proportional the stick value of the input channel. In limit values of the input channel (+/-125) the output channel moves in one second until it stops.
Incstop	Same as "Inc". In addition, on reaching the middle position of the output channel, the output channel remains standing on the center position. Only after the input channel has been brought to the center position, then the output channel is moved away from the center position.
Rev	Reverses the direction of action of the output channel.
Offset,Value%	The %offset is added to the output channel
Travel,Min%,Max%	The output channel is multiplied linearly with the specified percentage. Min / Max are used for separate adjustment of negative and positive effective range. When Max% is missing then Max% = Min%
Limit,Min%,Max%	If the output channel exceeds the specified limits, then the output channel is limited to the given value. Min / Max are used for separate adjustment of negative and positive effective range. When Max% is missing then Max% = Min%
Expo,Value%	The output channel is evaluated by an exponential function with the specified value (in %). Values are ±100%.with 0% = no effect.
Switch, Threshold**%,Min%,Max%**	If the output exceeds the specified threshold (in %), then the output is set to Max%. Otherwise, the output is set to Min%. Giving each a percentage of max. Range of the output channels
Curve,P1%,P2%,..,Pn%	The output channel is evaluated with a curve from points P1 to Pn%%. The points are specified as a percentage of the maximum range. At least two points must be specified, there can be any number of them. The points are distributed equally spaced over the full range of the output channel.

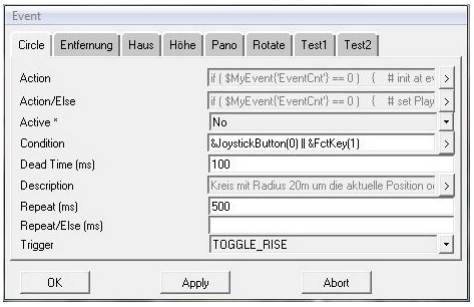
The functions are evaluated to the output channels in the following order:

1. Multi-point curve (Curve)
2. Expo (Expo)
3. Linear Travel Limiting (Travel)
4. Switch (Switch)
5. Reverse (Rev)
6. Limiter (Limit)

The data for each output channel and output parameters are separated by "+" from each other.

Example	Inputted Data
Incremental Poti, composed of two joystick POV-Buttons	`inc + JoystickPov90,0,40 + JoystickPov270,0,-40`
Control of External Control Yaw via the serial channel 11 from the event "Control Gier" and also through a joystick. With linear displacement limit and expo for the joystick, so a very sensitive yaw is possible.	`SerialChannel11 + JoystickR,30,30,35`
Setting a imagination-curve for the output channel	`JoystickX + Curve,-50,40,90,22,-100,100`

21. Event Engine.



With the **Event Engine** certain actions can take place depending on occurring Events. The event engine has access to all internal variables, functions, and controls of Mission Cockpit. This makes it very powerful.

The settings dialog is accessed via:

`File -> Event Configuration`

An event consists of the following components:

Component	Description	
Active	**Yes**	Event will be considered. The event name is displayed in the bottom status line.
	No	Event is not included. The event name is not shown in the bottom status line.
Condition	If the condition is **`true`** can trigger the event. Consists of one or more Perl statements.	
Action	Action that is executed when the event was triggered. Consists of one or more Perl statements.	
Action/Else	Action that is executed when the event is not triggered. Consists of one or more Perl statements.	
Idle Time	Time (in milliseconds) after execution of "action" until the next "action" can be executed.	
Repeat (ms)	Sets the cycle time to repeat the action command, if the event is active. Value should be a multiple of 50 ms (the internal cycle time of the event engine). If the field is empty or <50, the action command is started by 50ms/20Hz.	
Repeat/Else (ms)	As Repeat, but to Action/Else.	
Instruction	Descriptive text.	
Trigger	Specifies how to trigger the event.	
	RISE	The event will be triggered once when the condition of **`false`** to **`true`** changes.
	EVENT	The event will be triggered once when the condition of **`true`** to **`false`** changes.
	TOGGLE RISE	The event is triggered when the condition of **`false`** to **`true`** changes. The event will be terminated when the condition again by **`false`** to **`true`** changes.
	TOGGLE FALL	The event is triggered when the condition of **`true`** to **`false`** changes. The event will be terminated when the condition again of **`true`** to **`false`** changes.
	TRUE	The event is always triggered when the condition is **`true`**
	FALSE	The event is always triggered when the condition is **`false`** is

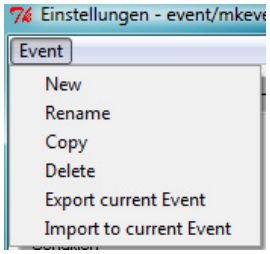
In principle, any number of events can be defined. The number is limited by the on screen available space of the configuration dialog.

Important:
For multi-line entries, the field is grayed out and not ready for input.

Multi-line fields can be edited with the editor that appears after clicking on the ">" symbol

Note:
After configuring the event a restart is not required. The events are now on the "**Apply**" or "**OK**" active. It can even the Perl instructions (Condition, Action, Action/Else) be changed during operation.

Note:
In menu "**Event**" can the events be created, deleted, copied or renamed. If you want to store a single event or reload it then use the import and export function.



Note:
Events are stored in the directory **`event/mkevent.xml`**. This can be changed, if needed, in the configuration of Startup Script

21.1. Perl Statements.

In the event fields "condition", "action" and "Action / Else" Perl instructions are executed. Simple things, without the knowledge of Perl language can be set. (See examples below). For more complex events, however, Perl skills and knowledge of the internal functioning of the Mission Cockpit are essential.

Important:
The instructions "Condition, Action, Action/Else" with high frequency (20 Hz or repeat frequency set) is executed repeatedly. The instructions should have a short processing time and, under no circumstances, may not block. Otherwise, the entire timing of Mission Cockpit will be messed up.

Important:
The Perl statements are recompiled every time they are used / executed. Too many and complex events, can increase the CPU load. Monitoring CPU utilization cannot hurt. Simple events hardly affect the CPU.

Important:
Syntax errors cannot harm the main program. Compiler errors are displayed in the CMD shell launched from the Mission Cockpit.

Important:
The Perl statements run in the context of the main program. This gives you access to all variables and functions. Already used by the Mission cockpit global variables can only be changed if you know exactly what you're doing.

Important:
If global variables are to be used, e.g. to have data from an event-run again available in the next event-run, should the variable names begin with a unique prefix.

Note:
There is no need to be "one-liners." Can be used also extensive, multi-line Perl programs with many instructions.

Note:
User defined functions can be saved as *. Pl, be stored in the directory **`plugin`**. They are loaded automatically at startup and do not have to be compiled for each event iteration.

Note:
With the simulator and the debug display from debug menu, the events to be used can be very well tested, before they are released on the flying MikroKopter.

A few simple examples:

Condition	`&JoystickButton(0)`	Returns "1" (= true), if the joystick button 1 is pressed
Condition	`&MouseButton(0)`	Returns "1" (= true), if the 3D-Mouse button 1 is pressed
Condition	`$MkOsd{UBat} < 10`	Returns "1" (= true) if the battery voltage <10 volts Here is accessed on the OSD dataset, can also be displayed in the Debug menu, "NC OSD record (O)"
Condition	`&Altitude() > 50`	Function &Altitude() returns an average from GPS altitude and barometric pressure height. Corresponds to the value that also appears in the OSD
Action	`&CbPlayerNext()`	Continue to next waypoint
Action	`&CbPlayerHome()`	Fly to Home Position
Action	`&SerialChannel(0, 50)`	Sets the serial channel "0" to the value "50"
Action	`&TtsSpeak('HIGH', "Hallo")`	Voice message "Hello" with high priority
Action	`&TtsSpeak('HIGH', $MkOsd{UBat})`	Voice announcement of the battery voltage with high priority

21.2. Events Examples.

With the program download are a few sample events. The events are disabled and require activation before use.

⚠ WARNING

The events in part by External access control grab directly the flight control of the MK. Before activation read the notice in the event description field, and Chapter 4.20 of "External Control"!

The description / documentation of the event displays by clicking on the ">" symbol on the right side

Event	Description
Circle	Flies in a circle of radius 20m. Circle center point is the current MK-position. If the POI mode is active, then the POI is the center of the circle. Mission Cockpit switches in the player paused mode, calculate and send the GPS target coordinates.
Distance	The distance to the home position is regularly announced by voice output.
House	Selected at random to one of the 44 possibilities, draws the "Home of Santa Claus", calculate the waypoints and flies from them. The edge length is 30m from the house. The house is built into the current MK-sight.
Panorama	Camera panorama function with shutter release via serial channel. For details see description of the event.
Rotate	Yaw with a defined speed.

21.3. Important Variables & Functions.

The following list is a small selection of the most important variables and functions that can be used at the events.

The program is constantly evolving and changing. I'm trying to avoid, but it can happen that some function calls may be changed.

Variable		Description

`$MkOsd{'Version'}`
`$MkOsd{'CurPos_Lon'}`
`$MkOsd{'CurPos_Lat'}`
`$MkOsd{'CurPos_Alt'}`
`$MkOsd{'CurPos_Stat'}`
`$MkOsd{'TargetPos_Lon'}`
`$MkOsd{'TargetPos_Lat'}`
`$MkOsd{'TargetPos_Alt'}`
`$MkOsd{'TargetPos_Stat'}`
`$MkOsd{'TargetPosDev_Dist'}`
`$MkOsd{'TargetPosDev_Bearing'}`
`$MkOsd{'HomePos_Lon'}`
`$MkOsd{'HomePos_Lat'}`
`$MkOsd{'HomePos_Alt'}`
`$MkOsd{'HomePos_Stat'}`
`$MkOsd{'HomePosDev_Dist'}`
`$MkOsd{'HomePosDev_Bearing'}`
`$MkOsd{'WaypointIndex'}`
`$MkOsd{'WaypointNumber'}`
`$MkOsd{'SatsInUse'}`
	`$MkOsd{'Altimeter'}`
	`$MkOsd{'Variometer'}`
	`$MkOsd{'FlyingTime'}`
	`$MkOsd{'UBat'}`
	`$MkOsd{'GroundSpeed'}`
	`$MkOsd{'Heading'}`
	`$MkOsd{'CompassHeading'}`
	`$MkOsd{'AngleNick'}`
	`$MkOsd{'AngleRoll'}`
	`$MkOsd{'RC_Quality'}`
	`$MkOsd{'MKFlags'}`
	`$MkOsd{'NCFlags'}`
	`$MkOsd{'Errorcode'}`
	`$MkOsd{'OperatingRadius'}`
	`$MkOsd{'TopSpeed'}`
	`$MkOsd{'TargetHoldTime'}`
	`$MkOsd{'MKFlags2'}`
	`$MkOsd{'SetPointAltitude'}`
	`$MkOsd{'Gas'}`
		Data from the NC OSD dataset.
		
		The data can also be displayed in the Debug menu

`$MkNcDebug{'Digital_00'}`
`$MkNcDebug{'Digital_01'}`
`$MkNcDebug{'Analog_00'}`
`$MkNcDebug{'Analog_01'}`
`$MkNcDebug{'Analog_02'}`
`$MkNcDebug{'Analog_03'}`
`$MkNcDebug{'Analog_04'}`
`$MkNcDebug{'Analog_05'}`
`$MkNcDebug{'Analog_06'}`
`$MkNcDebug{'Analog_07'}`
`$MkNcDebug{'Analog_08'}`
`$MkNcDebug{'Analog_09'}`
`$MkNcDebug{'Analog_10'}`
`$MkNcDebug{'Analog_11'}`
`$MkNcDebug{'Analog_12'}`
`$MkNcDebug{'Analog_13'}`
`$MkNcDebug{'Analog_14'}`
	`$MkNcDebug{'Analog_15'}`
	`$MkNcDebug{'Analog_16'}`
	`$MkNcDebug{'Analog_17'}`
	`$MkNcDebug{'Analog_18'}`
	`$MkNcDebug{'Analog_19'}`
	`$MkNcDebug{'Analog_20'}`
	`$MkNcDebug{'Analog_21'}`
	`$MkNcDebug{'Analog_22'}`
	`$MkNcDebug{'Analog_23'}`
	`$MkNcDebug{'Analog_24'}`
	`$MkNcDebug{'Analog_25'}`
	`$MkNcDebug{'Analog_26'}`
	`$MkNcDebug{'Analog_27'}`
	`$MkNcDebug{'Analog_28'}`
	`$MkNcDebug{'Analog_29'}`
	`$MkNcDebug{'Analog_30'}`
	`$MkNcDebug{'Analog_31'}`
		Data from the NC data debug.
		
		The data can also be displayed in the Debug menu

`$Stick{'StickRange'}`
`$Stick{'JoystickX'}`
`$Stick{'JoystickY'}`
`$Stick{'JoystickZ'}`
`$Stick{'JoystickR'}`
`$Stick{'JoystickU'}`
`$Stick{'JoystickV'}`
`$Stick{'JoystickPov'}`
`$Stick{'Joystickutton'}`
	`$Stick{'MouseRotX'}`
	`$Stick{'MouseRotY'}`
	`$Stick{'MouseRotZ'}`
	`$Stick{'MouseTranX'}`
	`$Stick{'MouseTranY'}`
	`$Stick{'MouseTranZ'}`
	`$Stick{'MouseButton'}`
	`$Stick{'FctKey'}`
	`$Stick{'FctKeyToggle'}`
		Joystick, 3D-Mouse, Function keys, RC-Poti.
		
		The data can also be displayed in the Debug menu

`$Cfg->{'Reiter'}->{'Value'}`		Configuration

Functions	Description
&CbPlayerPlayPause()	Toggle: Play—Pause
&CbPlayerNext()	Next Waypoint
&CbPlayerPrev()	Previous Waypoint
&CbPlayerFirst()	First Waypoint
&CbPlayerLast()	Last Waypoint
&CbPlayerHome()	Home
&CbPlayerStop()	Stop Player
&CbPlayerWptKml()	Toggles between Waypoint und KML Mode.
&CbPlayerWptRandom()	Toggles between Random Player Modes: STD → RND → MAP
&CbPlayerRecord()	Toggles Recording on/off
&CbPlayerMute()	Toggles Voice Announcement on/off
&CbPoi()	Toggles POI-Display on/off
&CbGrid()	Toggles Grid on/off
&CbExit()	Quit Mission Cockpit
&JoystickButton(NUM)	Returns value of Joystick Button NUM
&MouseButton(NUM)	Returns value of 3D-Mouse Button NUM
&FctKey(NUM)	Returns value of Functions' Key NUM (1..12)
&FctKeyToggle(NUM)	Toggles between Function Key (1—12) press on and by next press off.
&CurPosIsValid()	check, if current GPS position is valid
&HomePosIsValid()	check, if home GPS position is valid
&TargetIsValid()	check, if target GPS position is valid
&MkIsMotorOn()	check, if motor are on
&MkIsFlying()	check, if MK is flying
&MkIsCalibrating()	check, if MK is calibrating
&MkIsMotorStarting()	check, if Motor is starting
&MkEmergencyLanding()	check, Emergency Landing
&MkIsFreeMode()	check, if MK is FREE Mode
&MkIsPhMode()	check, if MK is in PH Mode
&MkIsWptMode()	check, Range Limit
&MkTargetReached()	check, Target reached
&MkManualControl()	check, Manual Control
&AltitudeAir ()	Get altitude (Pressure Sensor)
&AltitudeGPS ()	Get altitude (GPS)
&Altitude ()	Get altitude (average Pressure Sensor , GPS)
&SerialChannel(Num, Val)	Set serial Channel value. Num: 0—11, Value Range: -128—0—127
&WpAdd(-x, -y, -lat, -lon, -alt)	Add a Waypoint x/y Pixel Coordinate
&WpDelete(n)	Delete Waypoint n
&WpLoadFile(filename)	Load Waypoints from file
&WpSaveFile(filename)	Save Waypoints to file
&WpSendAll()	Resend all Waypoints to MK
&TtsSpeak(Prio, Text)	TTS Voice Announcement Priority: LOW, MEDIUM, HIGH

22. Information for Flying Waypoints.

The MikroKopter must fly in Free-Mode properly. With little wind and Pressure Sensor switched on, MK may drift very little of.

The MikroKopter must in Free-Mode be safely controlled, before you start flying waypoints. The MikroKopter flies (almost) by itself, but can be in difficult situations, that you have to intervene, possibly even from great distances, to retrieve it safely and securely.

My experience shows that 6 satellites are not sufficient for a safe flight with waypoints. It should be 8 or more satellites received.

23. Licence.

Creative Commons license

You are free:

To Share — to copy, distribute and transmit the work.
To Remix — to adapt the work

Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Noncommercial — You may not use this work for commercial purposes.

Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

With the understanding that:

Waiver — Any of the above conditions can be **waived** if you get permission from the copyright holder

Public Domain — Where the work or any of its elements is in the **public domain** under applicable law, that status is in no way affected by the license.

Other Rights — In no way are any of the following rights affected by the license:

- Your fair dealing or **fair use** rights, or other applicable copyright exceptions and limitations;
- The author's **moral** rights;
- Rights other persons may have either in the work itself or in how the work is used, such as **publicity** or privacy rights

Notice — For any reuse or distribution, you must make clear to others the license terms of this work.

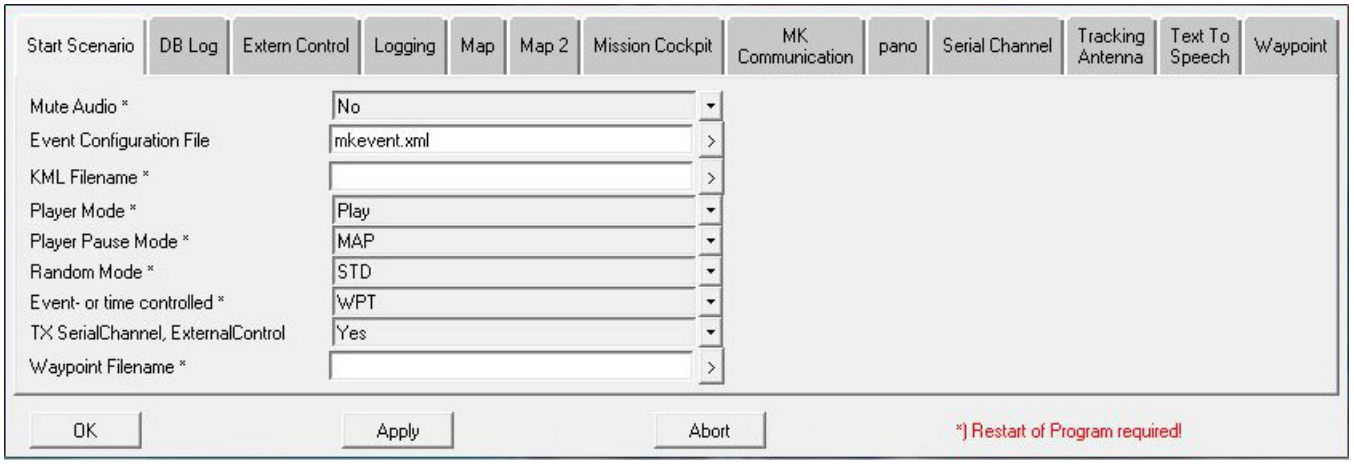
More details about the license here:
Short form: http://creativecommons.org/licenses/by-nc-sa/3.0/de/
Complete: http://creativecommons.org/licenses/by-nc-sa/3.0/de/legalcode

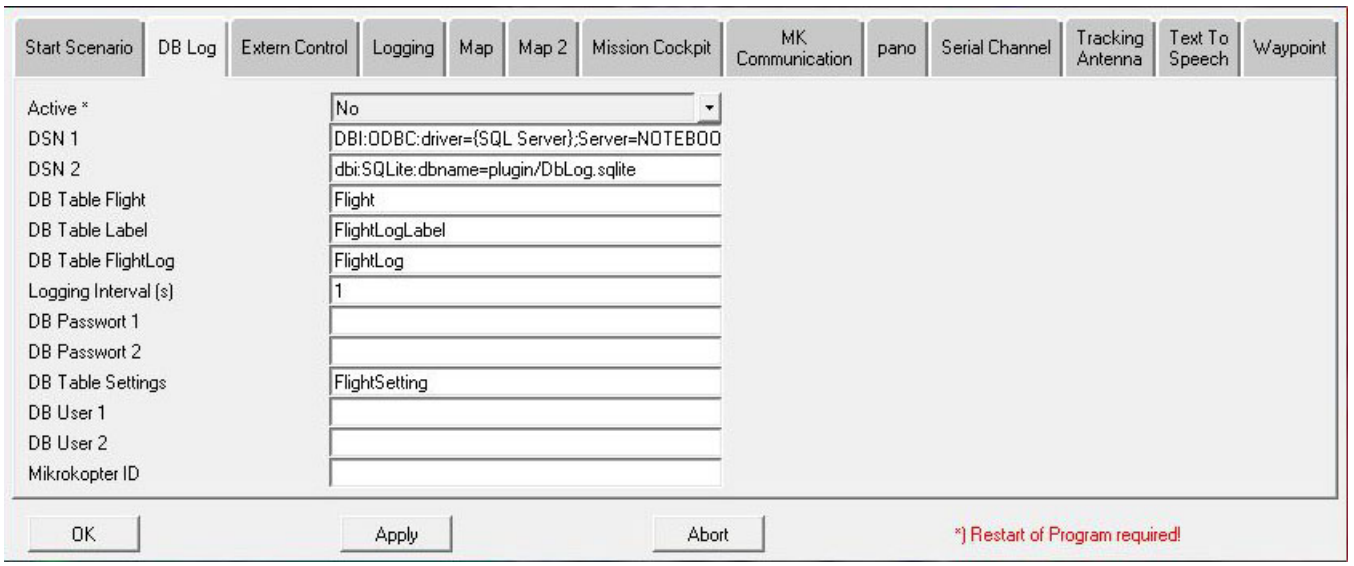
24. Mikrokopter Forum.

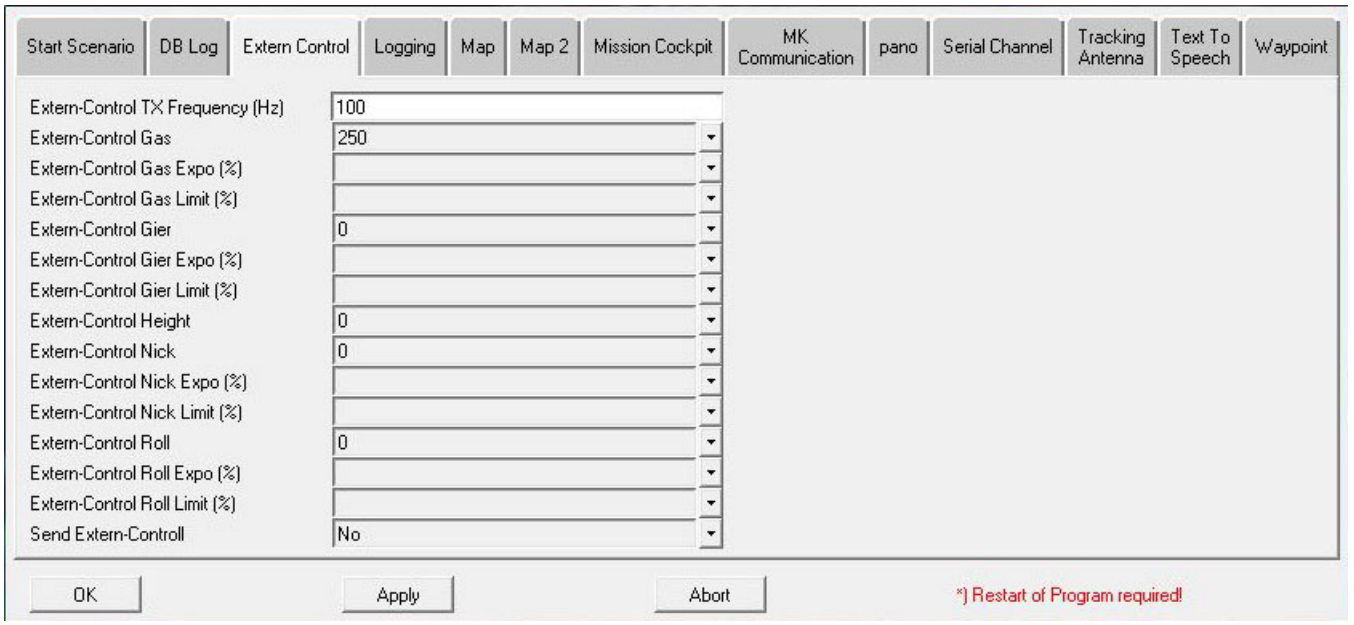
http://forum.mikrokopter.de/topic-8404.html
http://forum.mikrokopter.de/topic-28669.html

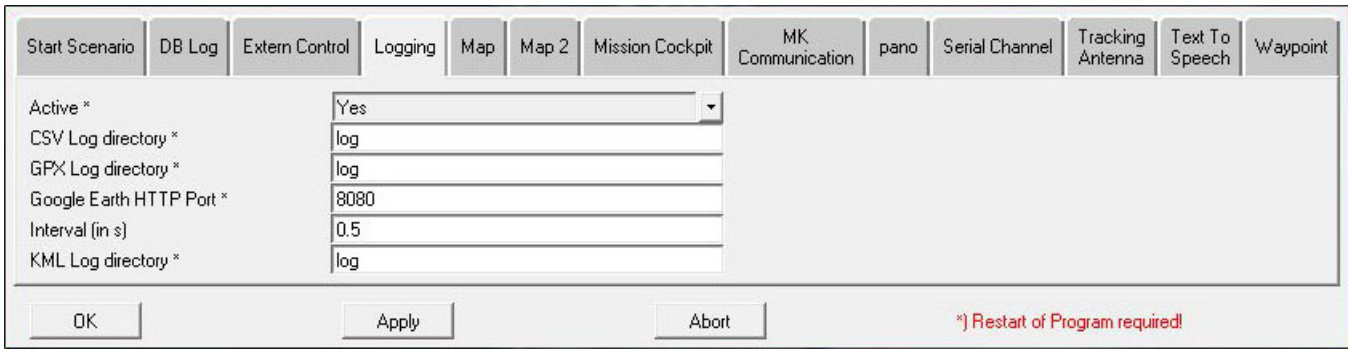
You are welcome to write feedback in the forum.

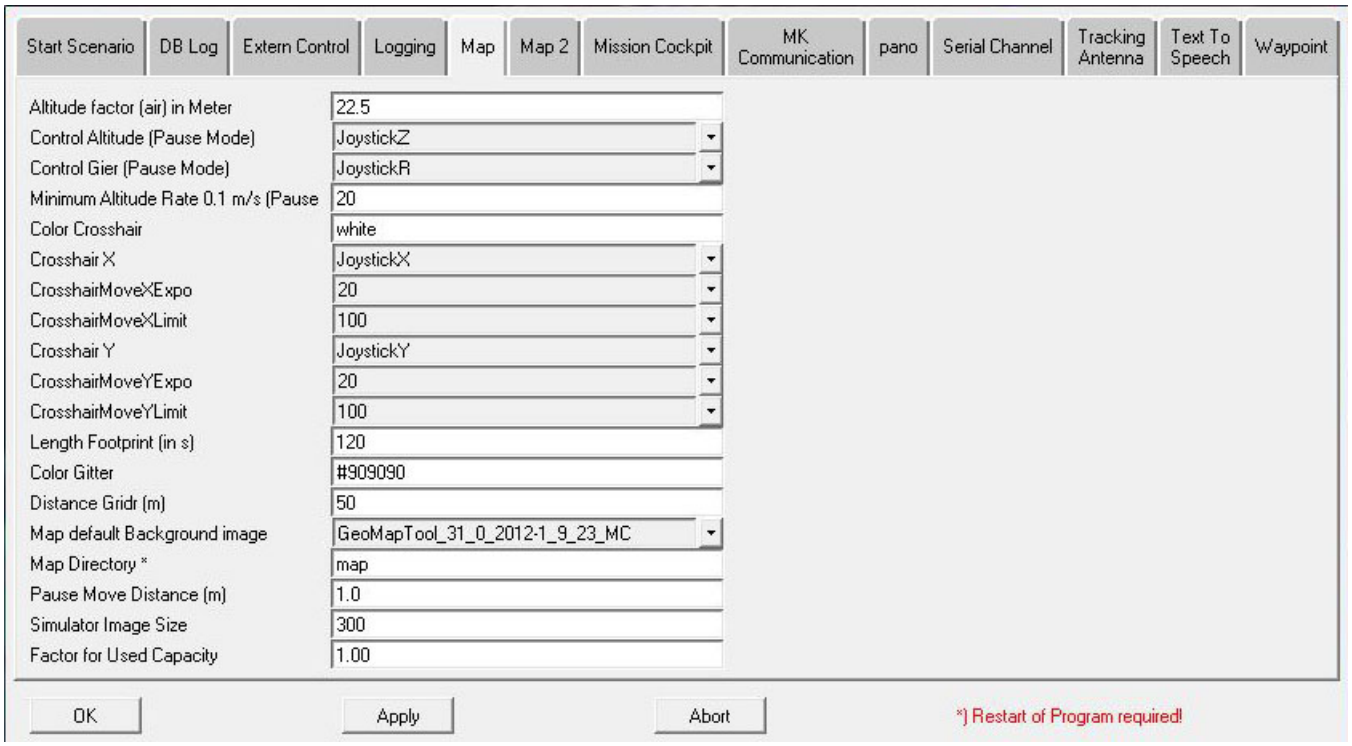
25. Appendix I.- Preferences.

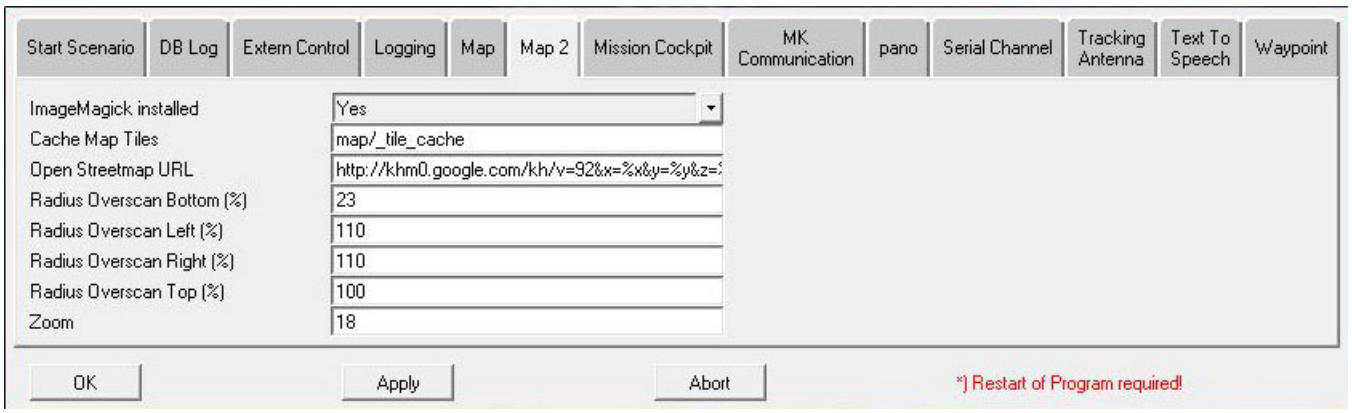


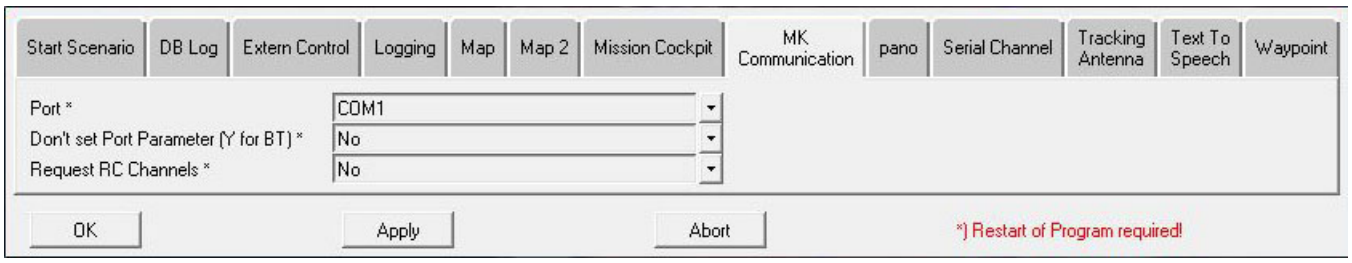


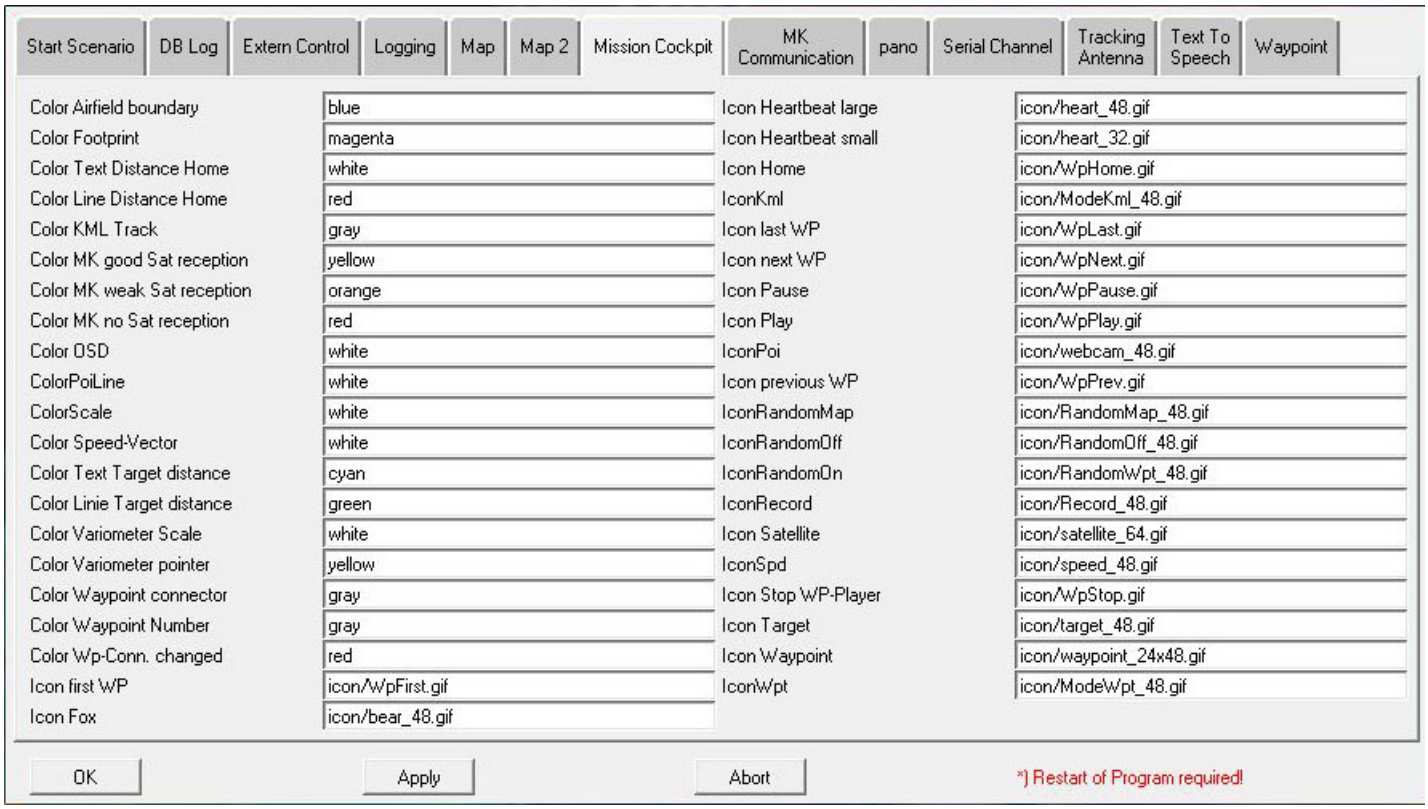


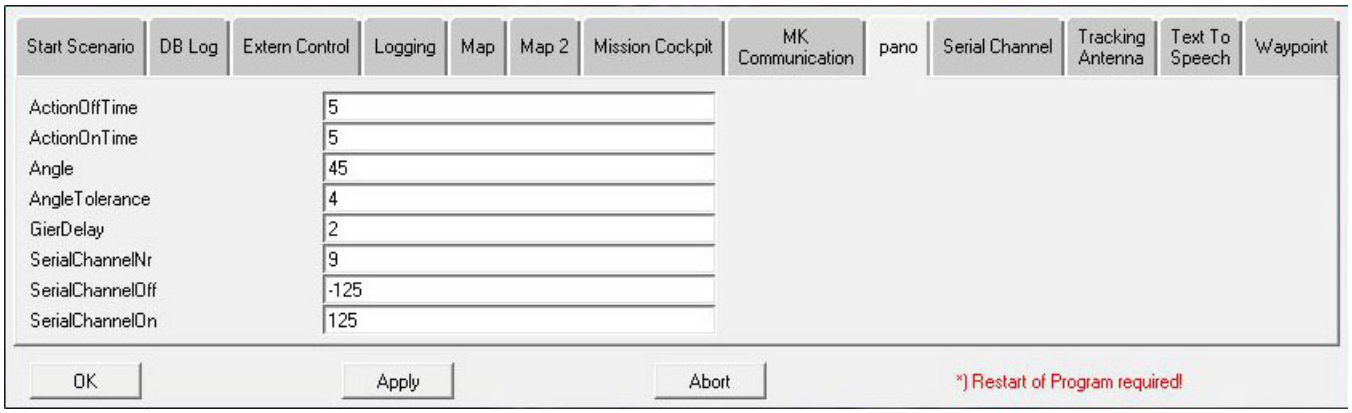

Start Scenario
DB Log
Extern Control
Logging
Map
Map 2
Mission Cockpit
MK Communication
pano
Serial Channel
Tracking Antenna
Text To Speech
Waypoint
Active *
Yes
CSV Log directory *
log
GPX Log directory *
log
Google Earth HTTP Port *
8080
Interval (in s)
0.5
KML Log directory *
log
OK
Apply
Abort
*) Restart of Program required!


Start Scenario
DB Log
Extern Control
Logging
Map
Map 2
Mission Cockpit
MK Communication
pano
Serial Channel
Tracking Antenna
Text To Speech
Waypoint
Altitude factor (air) in Meter
22.5
Control Altitude (Pause Mode)
JoystickZ
Control Gier (Pause Mode)
JoystickR
Minimum Altitude Rate 0.1 m/s (Pause
20
Color Crosshair
white
Crosshair X
JoystickX
CrosshairMoveXExpo
20
CrosshairMoveXLimit
100
Crosshair Y
JoystickY
CrosshairMoveYExpo
20
CrosshairMoveYLimit
100
Length Footprint (in s)
120
Color Gitter
#909090
Distance Gridr (m)
50
Map default Background image
GeoMapTool_31_0_2012-1_9_23_MC
Map Directory *
map
Pause Move Distance (m)
1.0
Simulator Image Size
300
Factor for Used Capacity
1.00
OK
Apply
Abort
*) Restart of Program required!

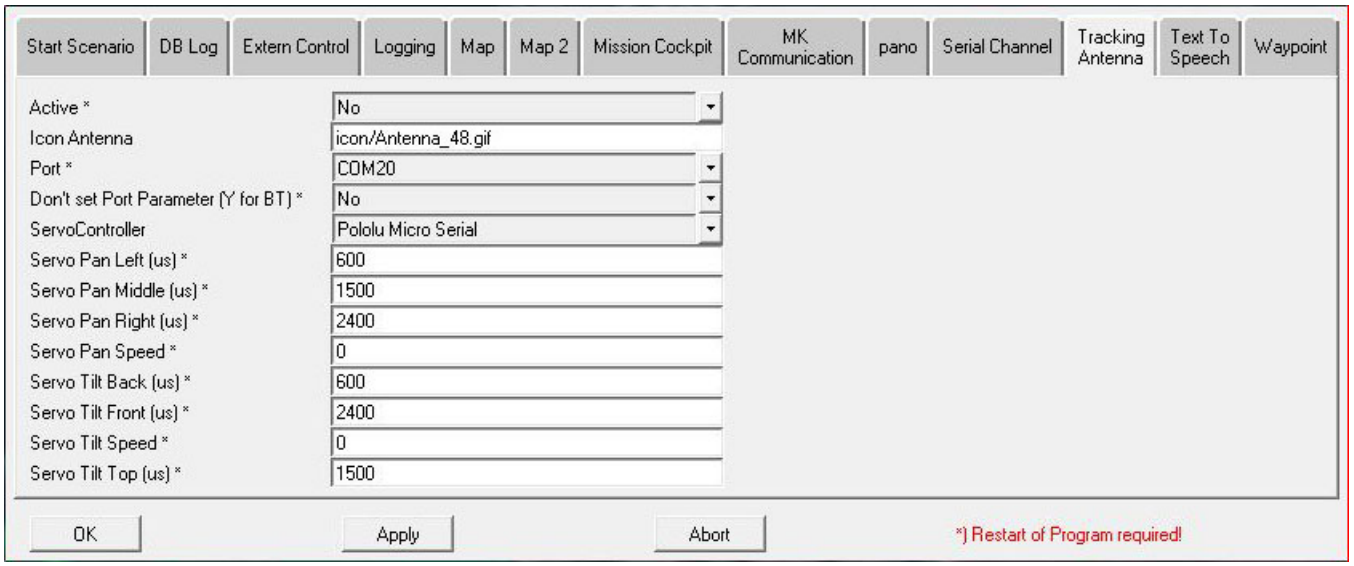

Start Scenario
DB Log
Extern Control
Logging
Map
Map 2
Mission Cockpit
MK Communication
pano
Serial Channel
Tracking Antenna
Text To Speech
Waypoint
ImageMagick installed
Yes
Cache Map Tiles
map/_tile_cache
Open Streetmap URL
http://khm0.google.com/kh/v=92&x=%x&y=%y&z=
Radius Overscan Bottom (%)
23
Radius Overscan Left (%)
110
Radius Overscan Right (%)
110
Radius Overscan Top (%)
100
Zoom
18
OK
Apply
Abort
*) Restart of Program required!

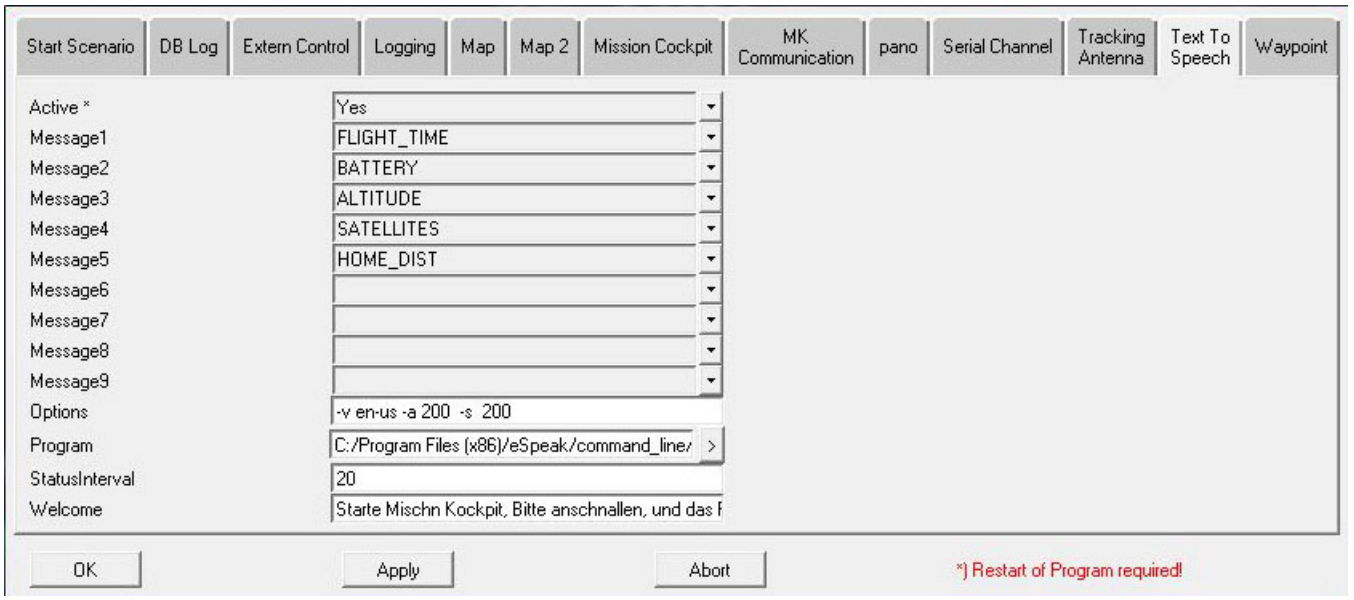

Start Scenario
DB Log
Extern Control
Logging
Map
Map 2
Mission Cockpit
MK Communication
pano
Serial Channel
Tracking Antenna
Text To Speech
Waypoint
Port *
COM1
Don't set Port Parameter (Y for BT) *
No
Request RC Channels *
No
OK
Apply
Abort
*) Restart of Program required!

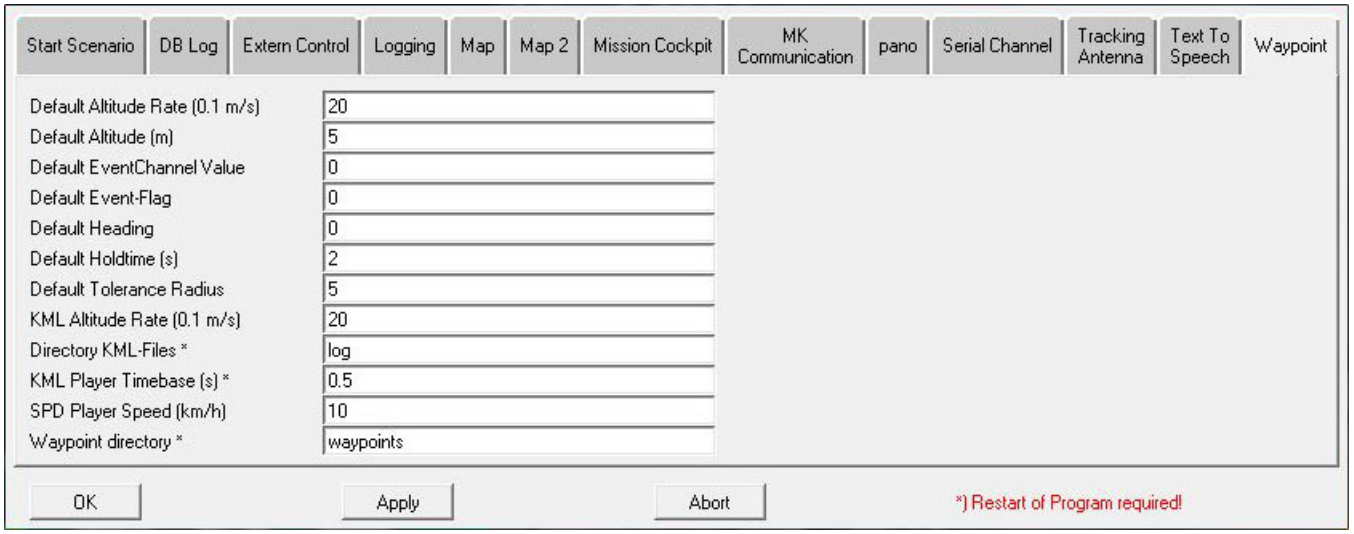

Start Scenario
DB Log
Extern Control
Logging
Map
Map 2
Mission Cockpit
MK Communication
pano
Serial Channel
Tracking Antenna
Text To Speech
Waypoint
Color Airfield boundary
blue
Color Footprint
magenta
Color Text Distance Home
white
Color Line Distance Home
red
Color KML Track
gray
Color MK good Sat reception
yellow
Color MK weak Sat reception
orange
Color MK no Sat reception
red
Color OSD
white
ColorPoiLine
white
ColorScale
white
Color Speed-Vector
white
Color Text Target distance
cyan
Color Linie Target distance
green
Color Variometer Scale
white
Color Variometer pointer
yellow
Color Waypoint connector
gray
Color Waypoint Number
gray
Color Wp-Conn. changed
red
Icon first WP
icon/WpFirst.gif
Icon Fox
icon/bear_48.gif
Icon Heartbeat large
icon/heart_48.gif
Icon Heartbeat small
icon/heart_32.gif
Icon Home
icon/WpHome.gif
IconKml
icon/ModeKml_48.gif
Icon last WP
icon/WpLast.gif
Icon next WP
icon/WpNext.gif
Icon Pause
icon/WpPause.gif
Icon Play
icon/WpPlay.gif
IconPoi
icon/webcam_48.gif
Icon previous WP
icon/WpPrev.gif
IconRandomMap
icon/RandomMap_48.gif
IconRandomOff
icon/RandomOff_48.gif
IconRandomOn
icon/RandomWpt_48.gif
IconRecord
icon/Record_48.gif
Icon Satellite
icon/satellite_64.gif
IconSpd
icon/speed_48.gif
Icon Stop WP-Player
icon/WpStop.gif
Icon Target
icon/target_48.gif
Icon Waypoint
icon/waypoint_24x48.gif
IconWpt
icon/ModeWpt_48.gif
OK
Apply
Abort
*) Restart of Program required!


Start Scenario
DB Log
Extern Control
Logging
Map
Map 2
Mission Cockpit
MK Communication
pano
Serial Channel
Tracking Antenna
Text To Speech
Waypoint
ActionOffTime
5
ActionOnTime
5
Angle
45
AngleTolerance
4
GierDelay
2
SerialChannelNr
9
SerialChannelOff
-125
SerialChannelOn
125
OK
Apply
Abort
*) Restart of Program required!


Start Scenario
DB Log
Extern Control
Logging
Map
Map 2
Mission Cockpit
MK Communication
pano
Serial Channel
Tracking Antenna
Text To Speech
Waypoint
Serial Channel 1
Serial Channel 2
Serial Channel 3
Serial Channel 4
Serial Channel 5
Serial Channel 6
Serial Channel 7
Serial Channel 8
Serial Channel 9
Serial Channel 10
Serial Channel 11
Serial Channel 12
Serial Channel TX Frequency (Hz)
20
Send Serial Channel
No
OK
Apply
Abort
*) Restart of Program required!


Start Scenario
DB Log
Extern Control
Logging
Map
Map 2
Mission Cockpit
MK Communication
pano
Serial Channel
Tracking Antenna
Text To Speech
Waypoint
Active *
No
Icon Antenna
icon/Antenna_48.gif
Port *
COM20
Don't set Port Parameter (Y for BT) *
No
ServoController
Pololu Micro Serial
Servo Pan Left (us) *
600
Servo Pan Middle (us) *
1500
Servo Pan Right (us) *
2400
Servo Pan Speed *
0
Servo Tilt Back (us) *
600
Servo Tilt Front (us) *
2400
Servo Tilt Speed *
0
Servo Tilt Top (us) *
1500
OK
Apply
Abort
*) Restart of Program required!


Start Scenario
DB Log
Extern Control
Logging
Map
Map 2
Mission Cockpit
MK Communication
pano
Serial Channel
Tracking Antenna
Text To Speech
Waypoint
Active *
Yes
Message1
FLIGHT_TIME
Message2
BATTERY
Message3
ALTITUDE
Message4
SATELLITES
Message5
HOME_DIST
Message6
Message7
Message8
Message9
Options
-v en-us -a 200 -s 200
Program
C:/Program Files (x86)/eSpeak/command_line/
StatusInterval
20
Welcome
Starte Mischn Kockpit, Bitte anschnallen, und das
OK
Apply
Abort
*) Restart of Program required!


Start Scenario
DB Log
Extern Control
Logging
Map
Map 2
Mission Cockpit
MK Communication
pano
Serial Channel
Tracking Antenna
Text To Speech
Waypoint
Default Altitude Rate (0.1 m/s)
20
Default Altitude (m)
5
Default EventChannel Value
0
Default Event-Flag
0
Default Heading
0
Default Holdtime (s)
2
Default Tolerance Radius
5
KML Altitude Rate (0.1 m/s)
20
Directory KML-Files *
log
KML Player Timebase (s) *
0.5
SPD Player Speed (km/h)
10
Waypoint directory *
waypoints
OK
Apply
Abort
*) Restart of Program required!